(12) UK Patent Application (19) GB (11) 2 279 531 (13) A

(43) Date of A Publication 04.01.1995

(21) Application No 9313063.1

(22) Date of Filing 24.06.1993

(71) Applicant(s)
Sony United Kingdom Limited

(Incorporated in the United Kingdom)

Sony House, South Street, STAINES, Middlesex,
TW18 4PF, United Kingdom

(72) Inventor(s)
Nicholas Ian Saunders
Shima Ravji Varsani
Martin Rex Dorricott

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA,
United Kingdom

(51) INT CL^6
H04N 5/14 , G06T 7/20

(52) UK CL (Edition N)
H4F FD10 FD30J FD30R FD30T3 FGM

(56) Documents Cited
GB 2261342 A GB 2259625 A GB 2256341 A

(58) Field of Search
UK CL (Edition L) H4F FEP FEX FGM FGXX
INT CL^5 H04N 5/14 7/01
Online databases: WPI

(54) Motion compensated image interpolation

(57) The motion compensated temporal interpolator is capable of determining when a background scene is covered or uncovered by a foreground object by projecting motion vectors determined for the output field onto temporally adjacent input fields (i/p1, i/p2) and detecting the number of times each input pixel is used as a source for the output pixel - covering corresponds to multiple pixel use in the following field while uncovering corresponds to multiple pixel use in the preceding field. The motion vectors for output pixels in a covered area can then be corrected by forward projecting vectors from the preceding frame pair whereas output pixels in an uncovered area are corrected by backward projecting vectors from the following pair.

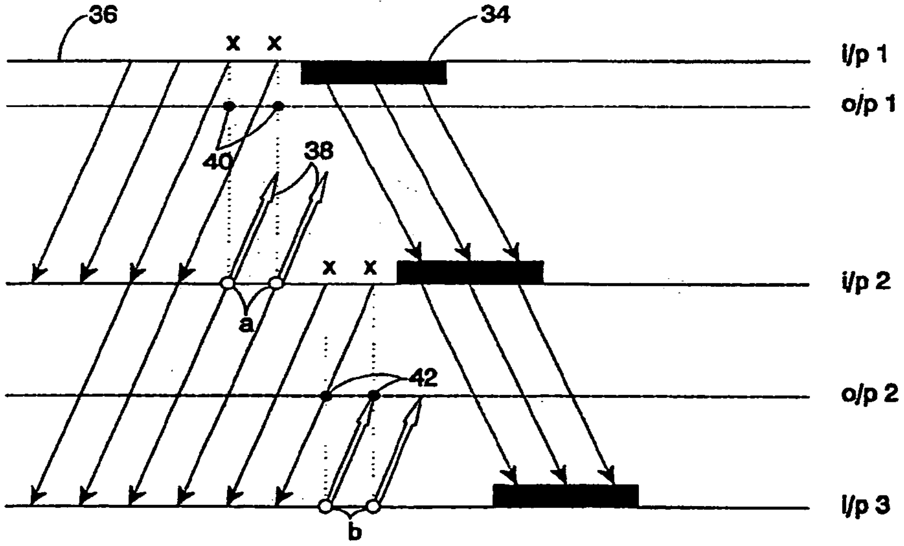


Fig.6

GB 2 279 531 A

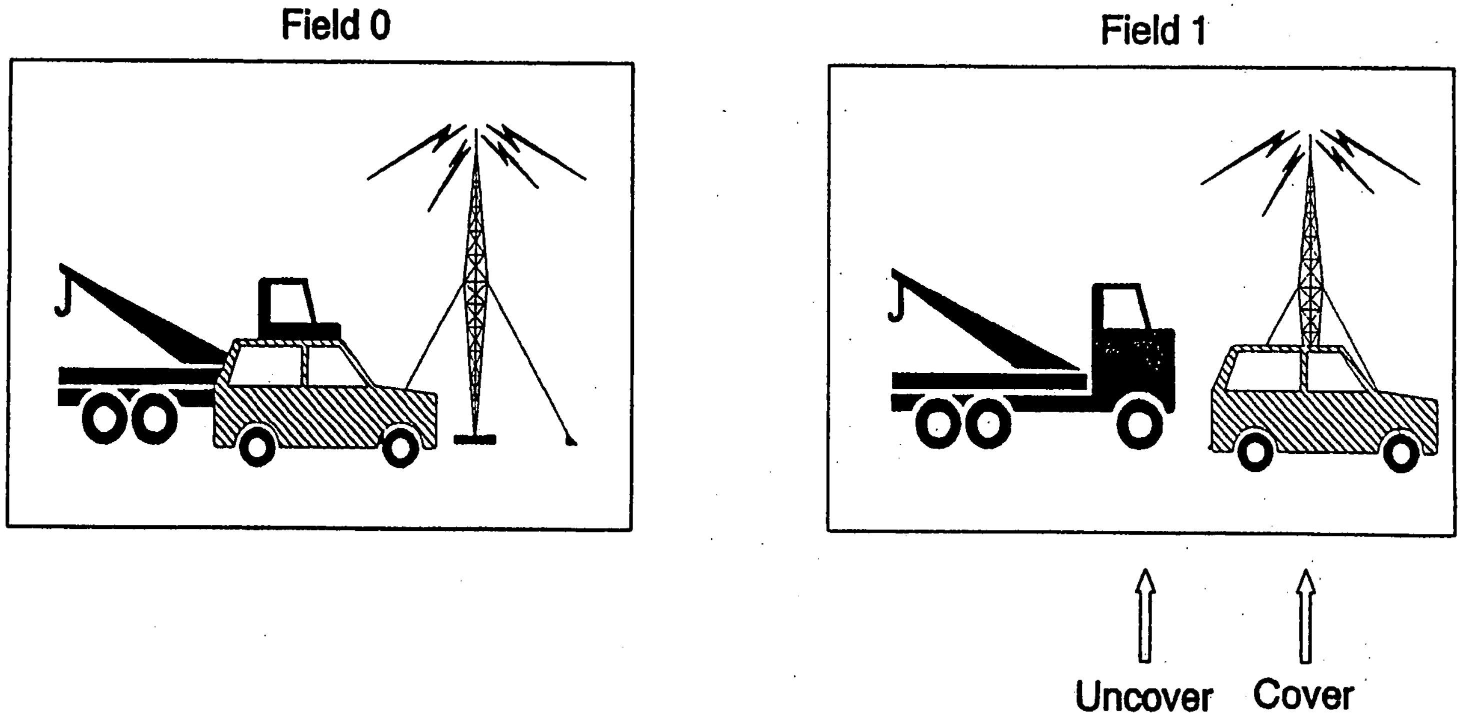


Fig.1

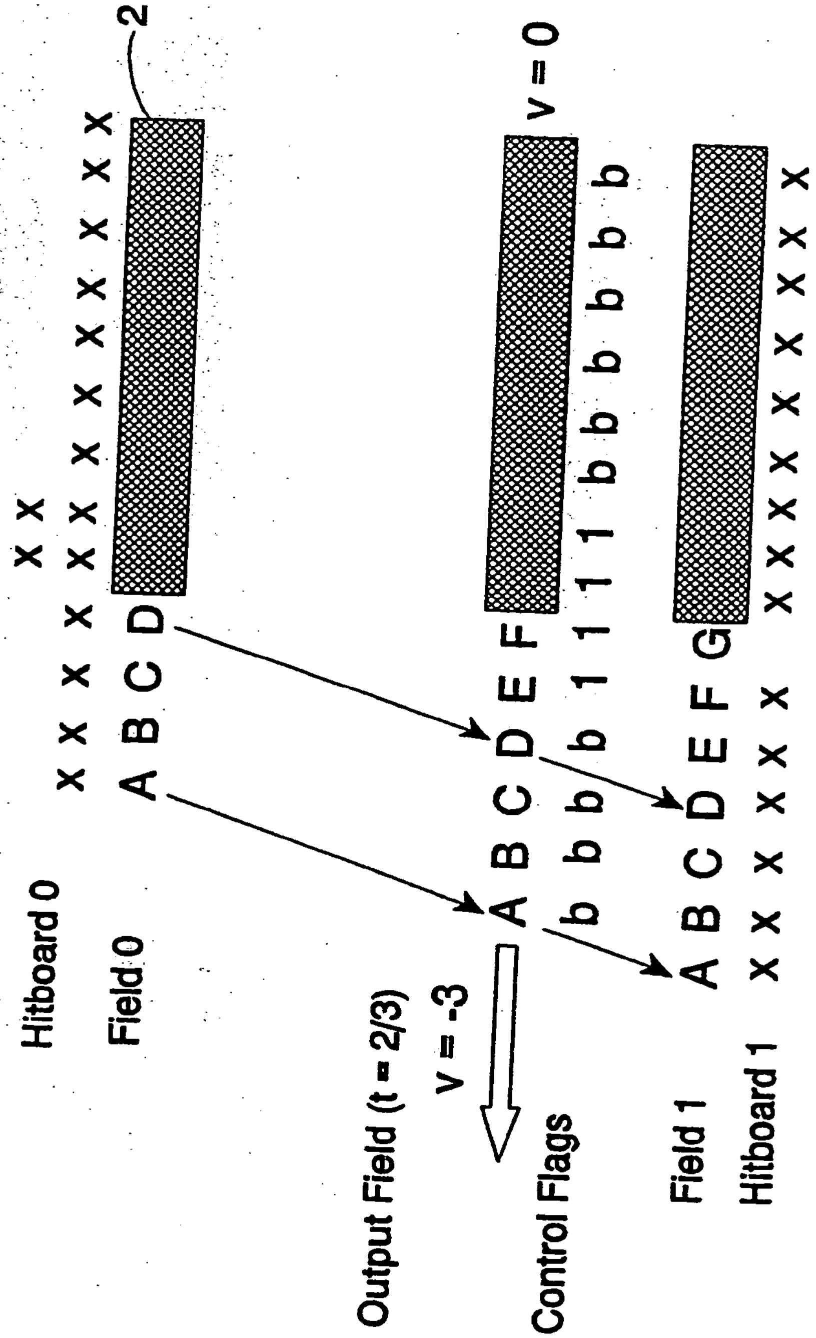
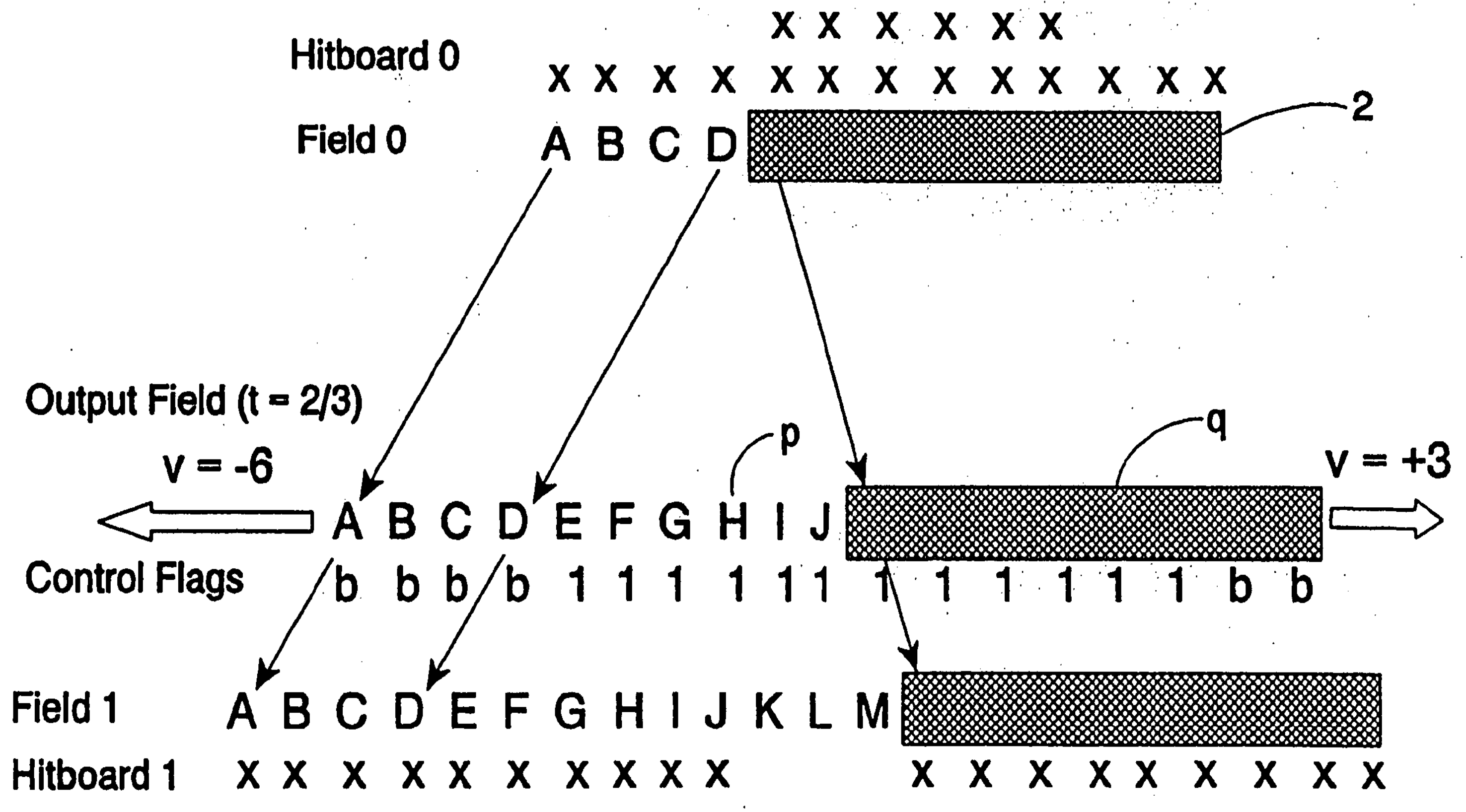

Hitboard 0
Field 0
x x x x x x x x x x x x x
x x
A B C D
2
Output Field (t = 2/3)
v = -3
A B C D E F
v = 0
Control Flags
b b b b 1 1 1 1 b b b b b b
Field 1
A B C D E F G
Hitboard 1
x x x x x x x x x x x x x x x x

Fig.2



Hitboard 0
X X X X X X
X X X X X X X X X X X X X
Field 0
A B C D
2
Output Field (t = 2/3)
v = -6
p
q
v = +3
A B C D E F G H I J
Control Flags
b b b b 1 1 1 1 1 1 1 1 1 1 1 1 b b
Field 1
A B C D E F G H I J K L M
Hitboard 1
X X X X X X X X X X X X X X X X X X X

Fig.3

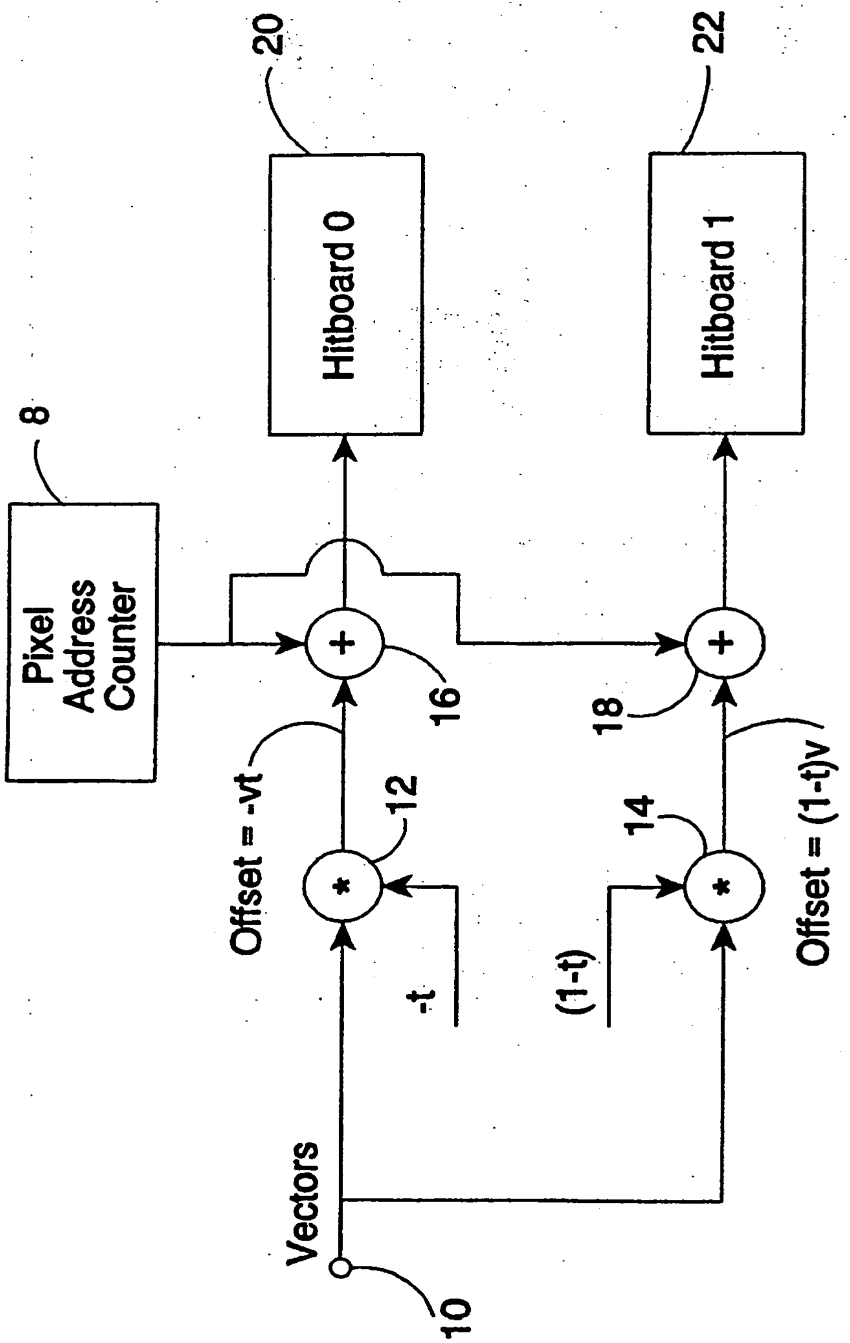

Vectors
10
*
Offset = -vt
12
-t
(1-t)
*
14
Offset = (1-t)v
Pixel Address Counter
8
+
16
+
18
Hitboard 0
20
Hitboard 1
22

Fig.4

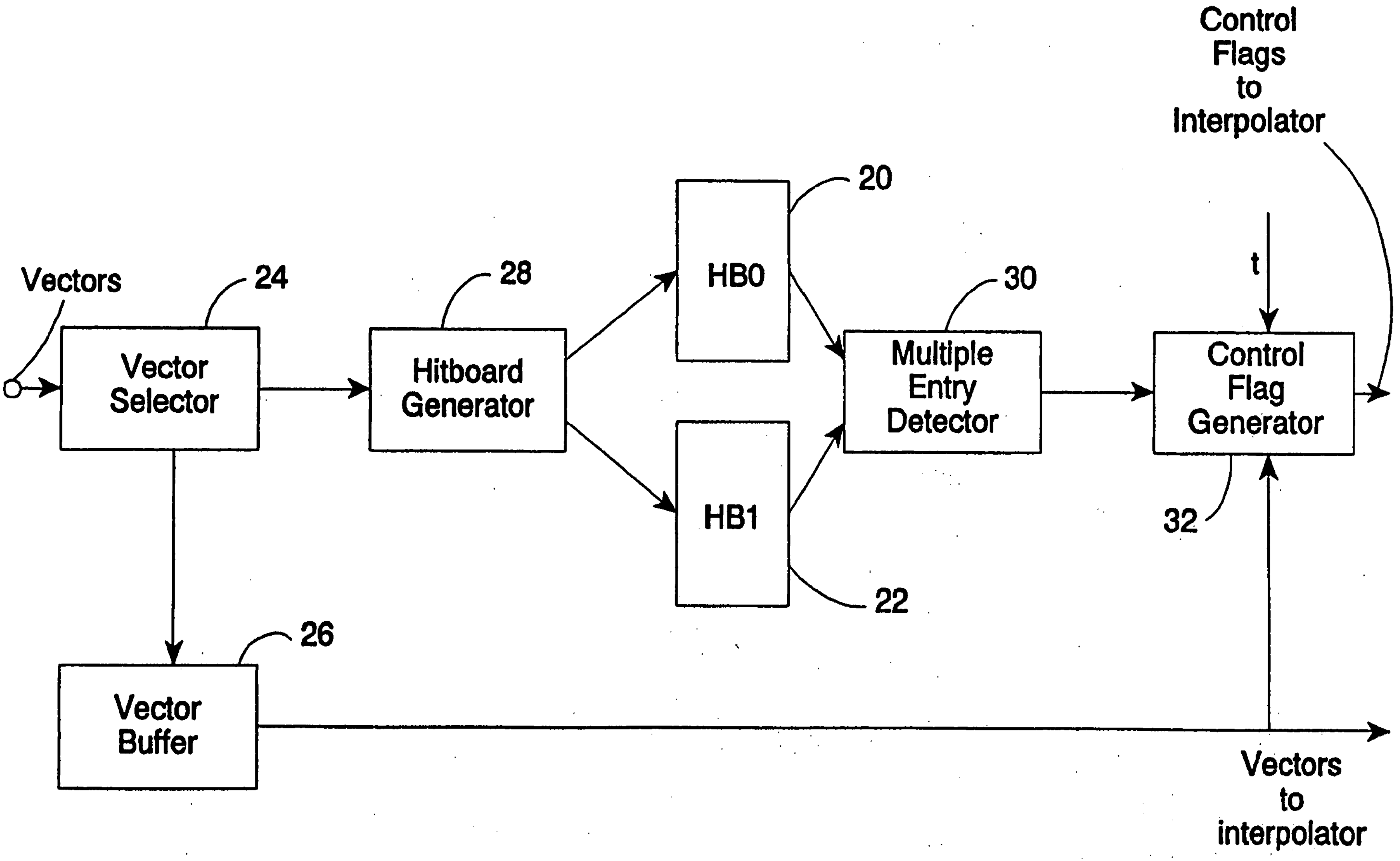

Control
Flags
to
Interpolator
Vectors
24
Vector
Selector
28
Hitboard
Generator
20
HB0
HB1
22
30
Multiple
Entry
Detector
t
Control
Flag
Generator
32
26
Vector
Buffer
Vectors
to
interpolator

Fig.5

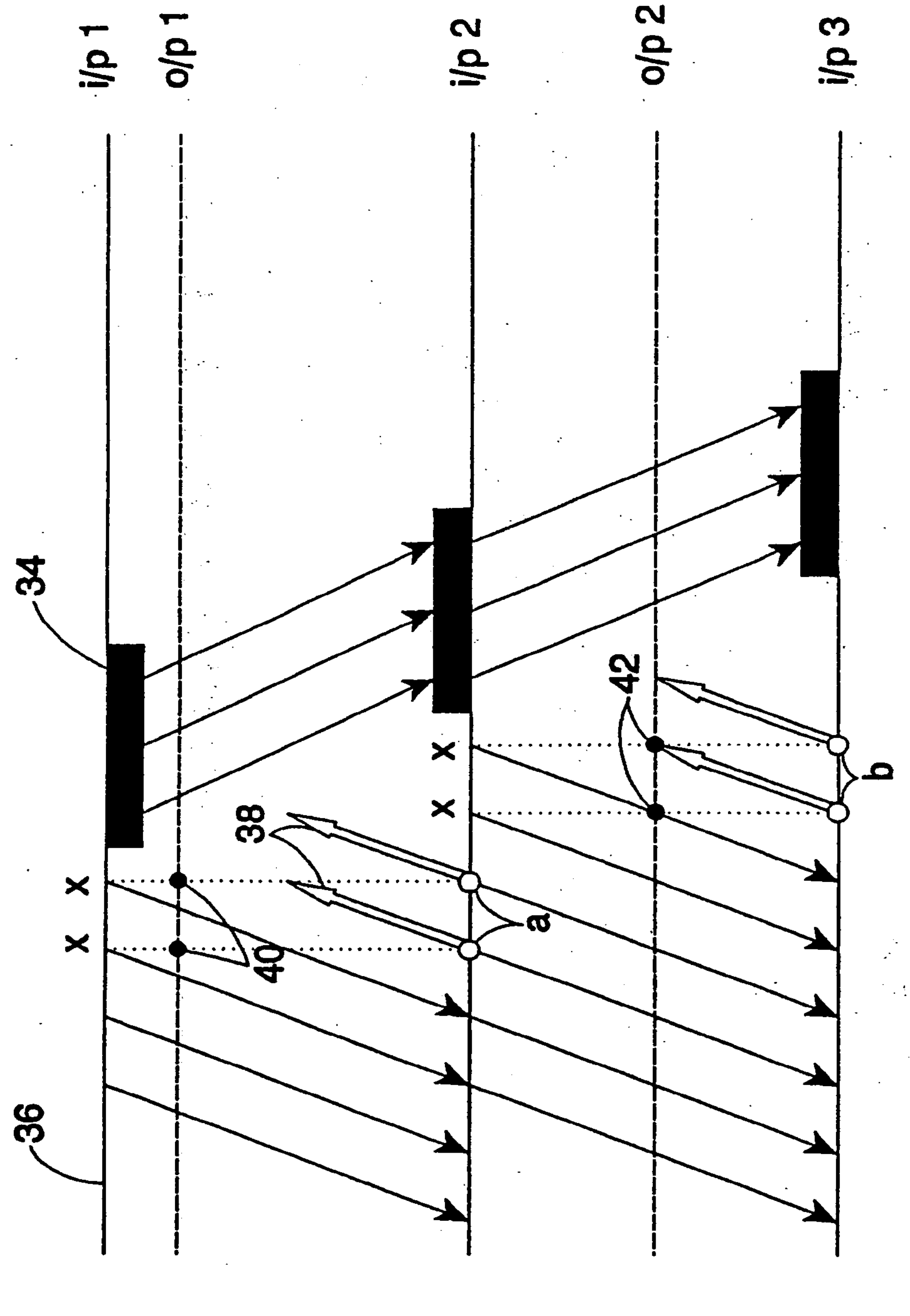


Fig.6

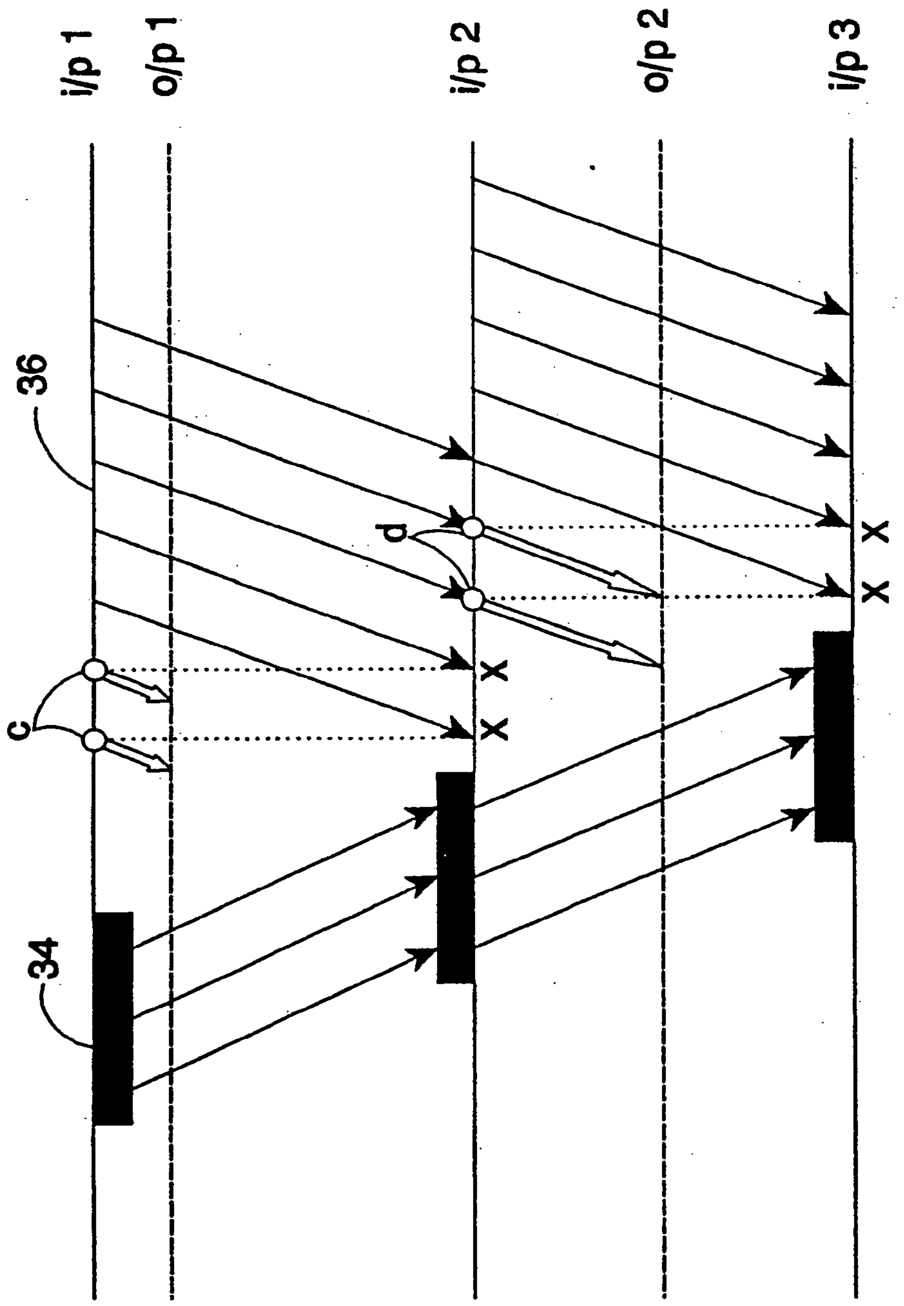


Fig.7

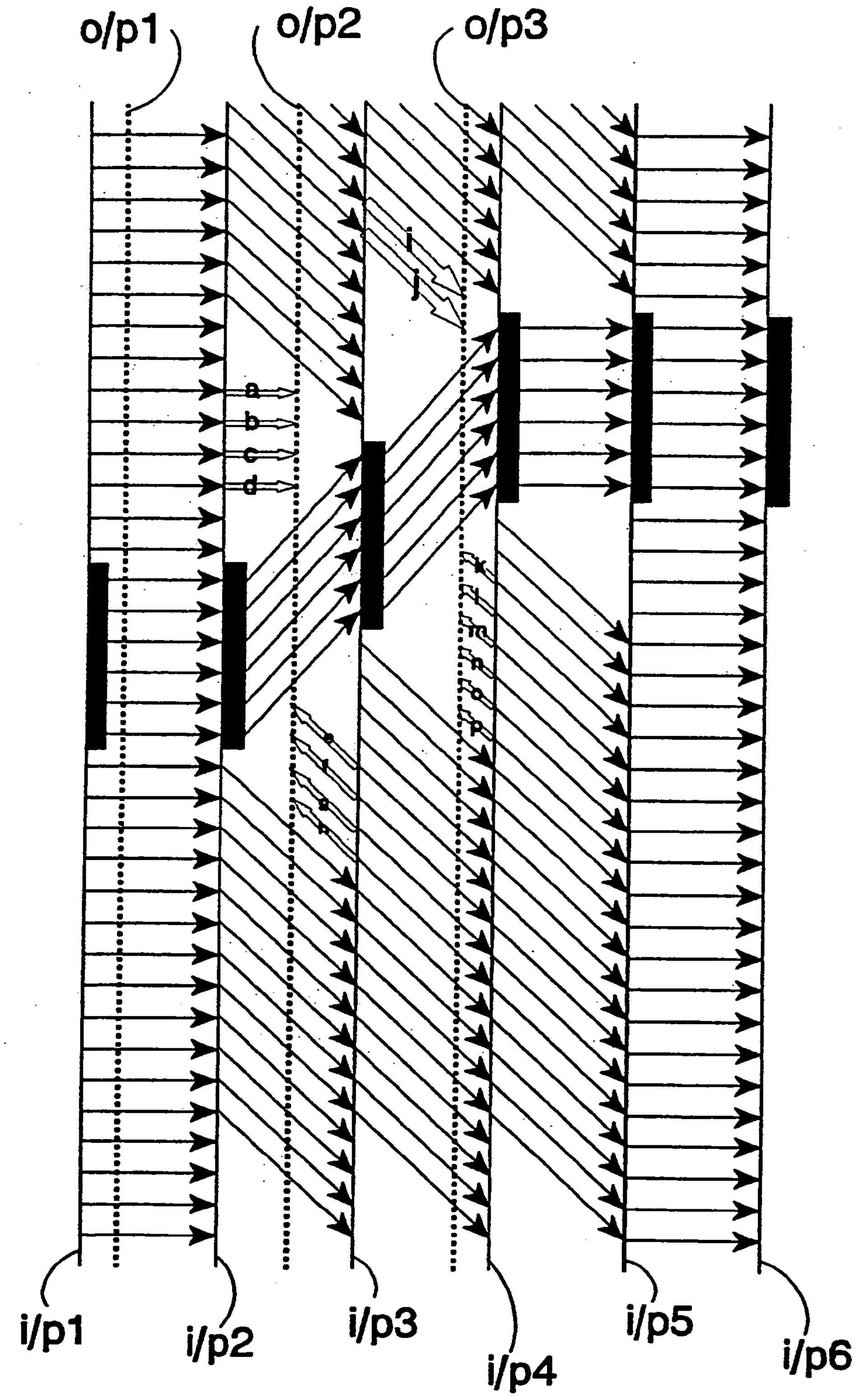


Fig.8

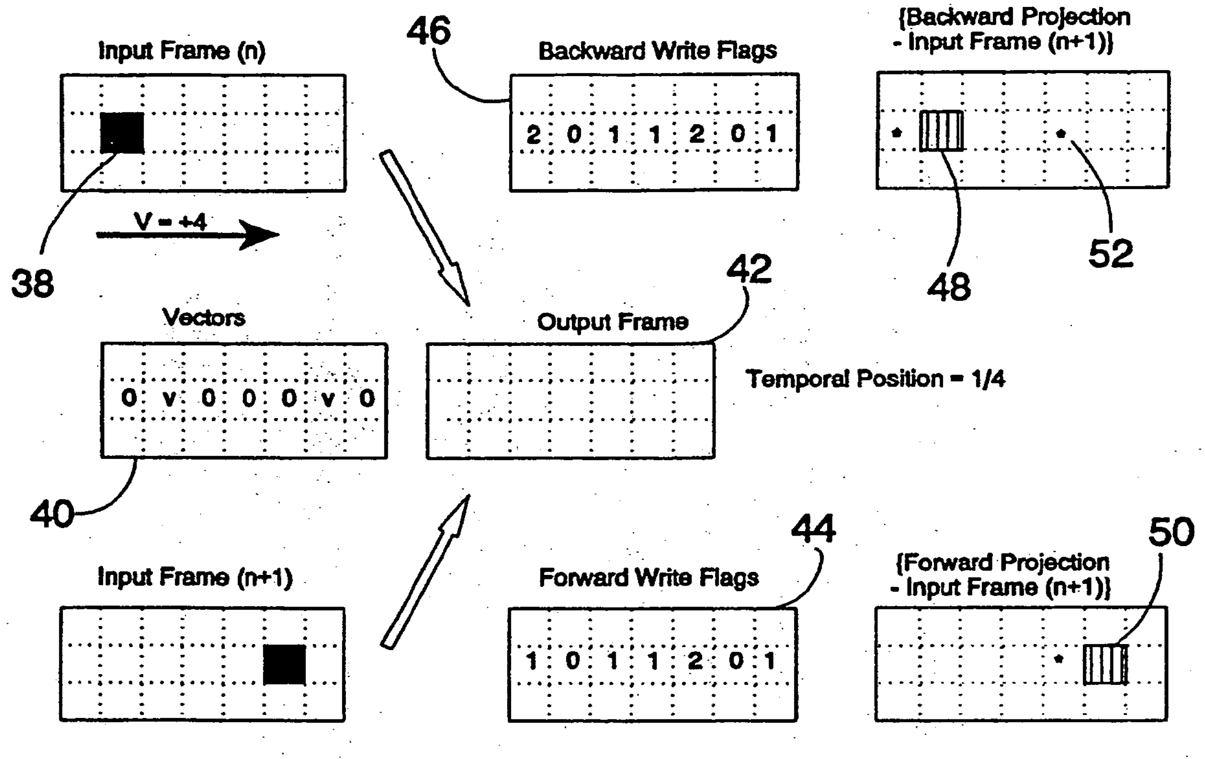

Input Frame (n)
46
Backward Write Flags
2 0 1 1 2 0 1
{Backward Projection
- Input Frame (n+1)}
V = +4
38
48
52
42
Vectors
0 v 0 0 0 v 0
Output Frame
Temporal Position = 1/4
40
44
50
Input Frame (n+1)
Forward Write Flags
1 0 1 1 2 0 1
{Forward Projection
- Input Frame (n+1)}

Fig.9

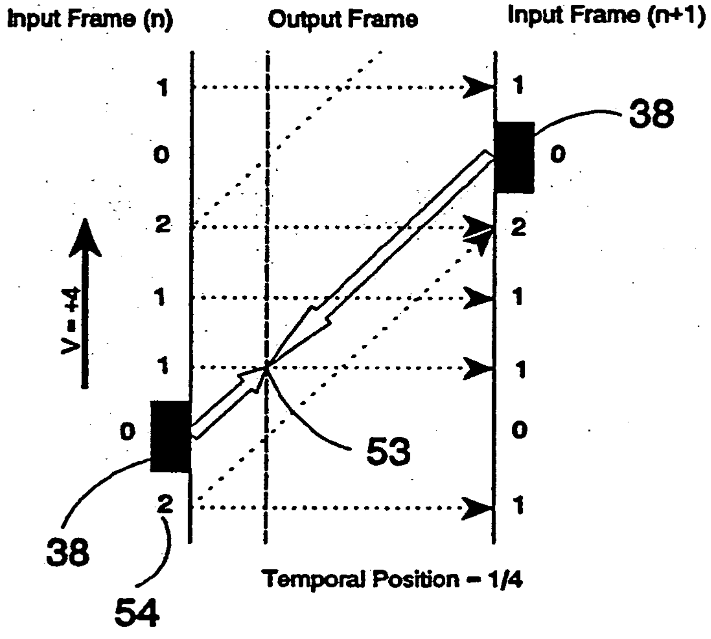

Input Frame (n)
Output Frame
Input Frame (n+1)
1
0
2
1
1
0
2
1
0
2
1
1
0
1
38
V = +4
53
38
54
Temporal Position = 1/4

Fig.10

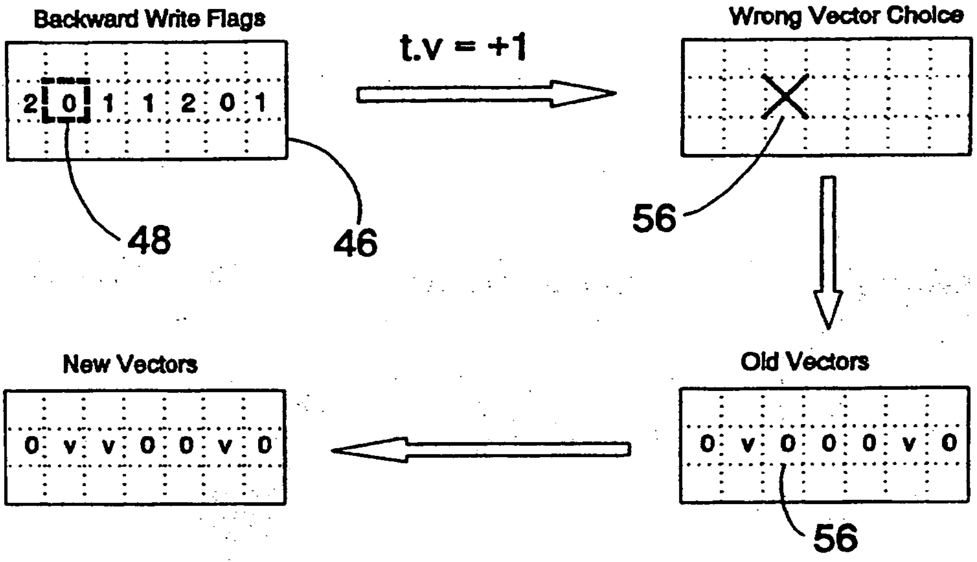

Backward Write Flags
2 0 1 1 2 0 1
48
46
t.v = +1
Wrong Vector Choice
56
Old Vectors
0 v 0 0 0 v 0
56
New Vectors
0 v v 0 0 v 0

Fig.11

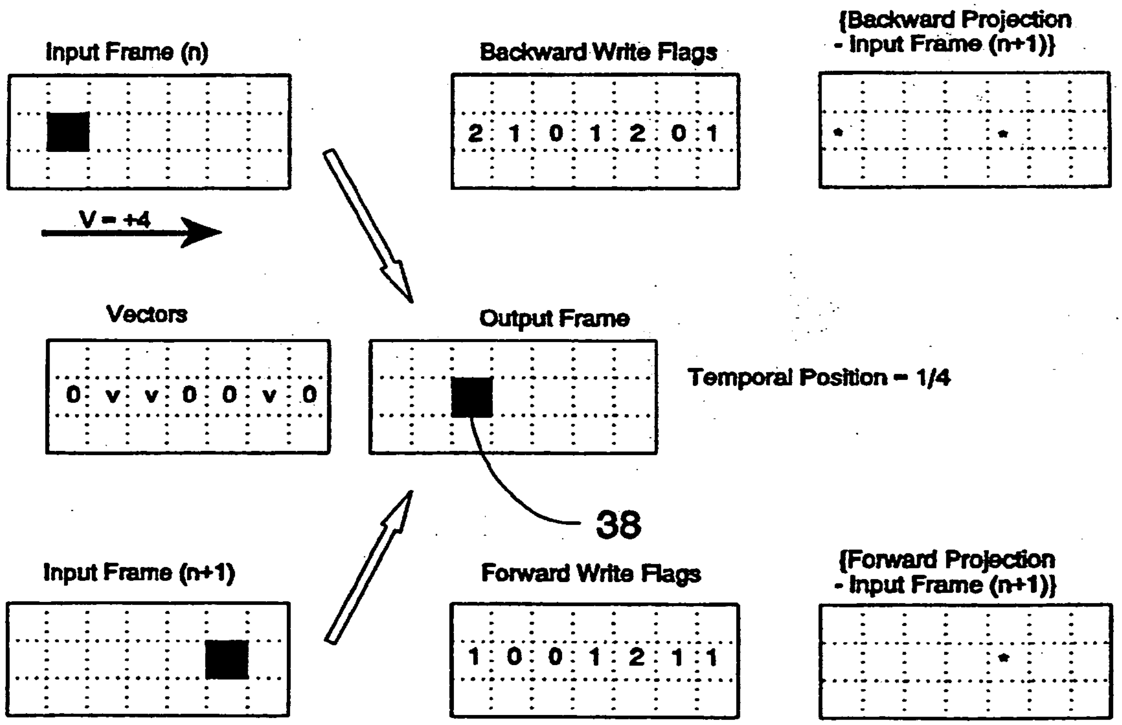
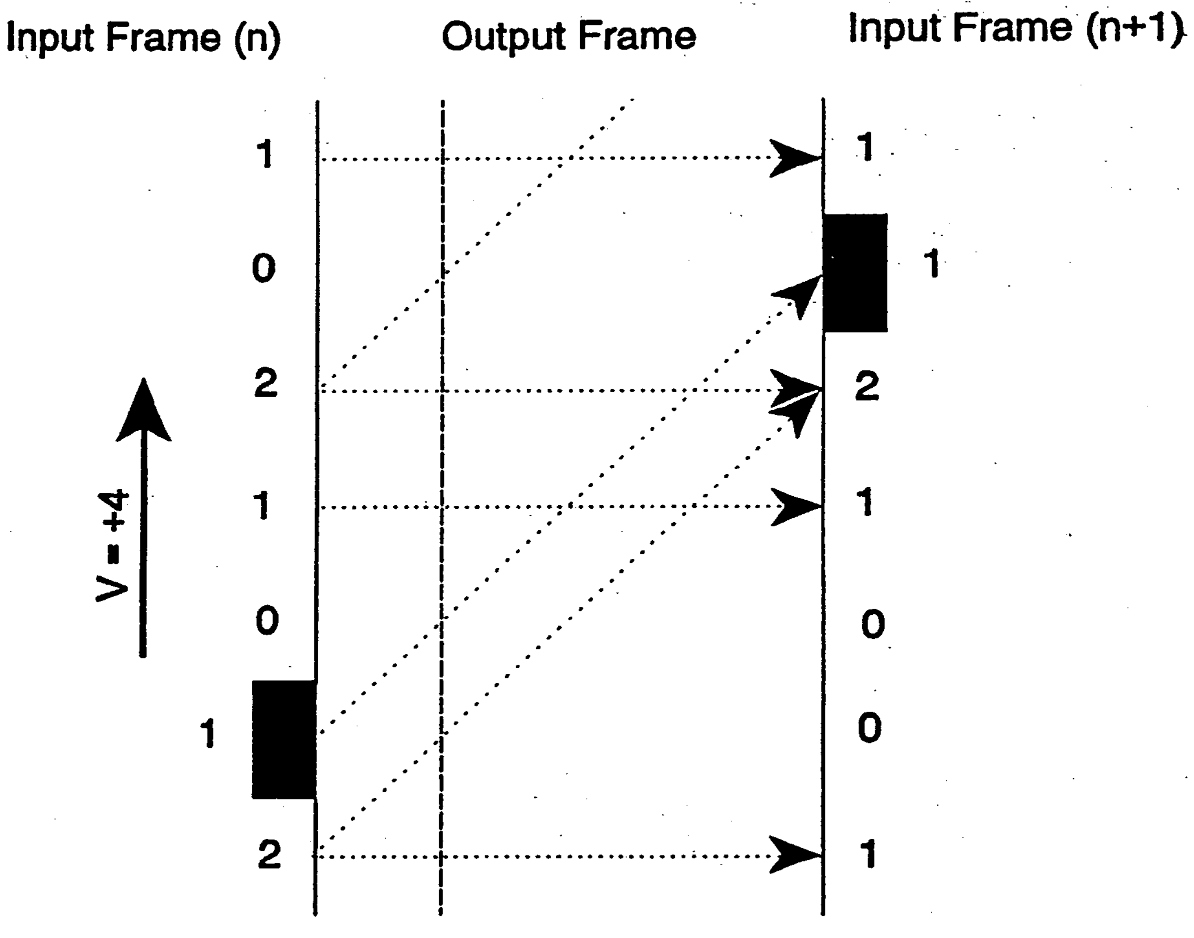

Input Frame (n)
V = +4
Backward Write Flags
2 1 0 1 2 0 1
{Backward Projection
- Input Frame (n+1)}
Vectors
0 v v 0 0 v 0
Output Frame
Temporal Position = 1/4
38
Input Frame (n+1)
Forward Write Flags
1 0 0 1 2 1 1
{Forward Projection
- Input Frame (n+1)}

Fig.12



Temporal Position = 1/4

Fig.13

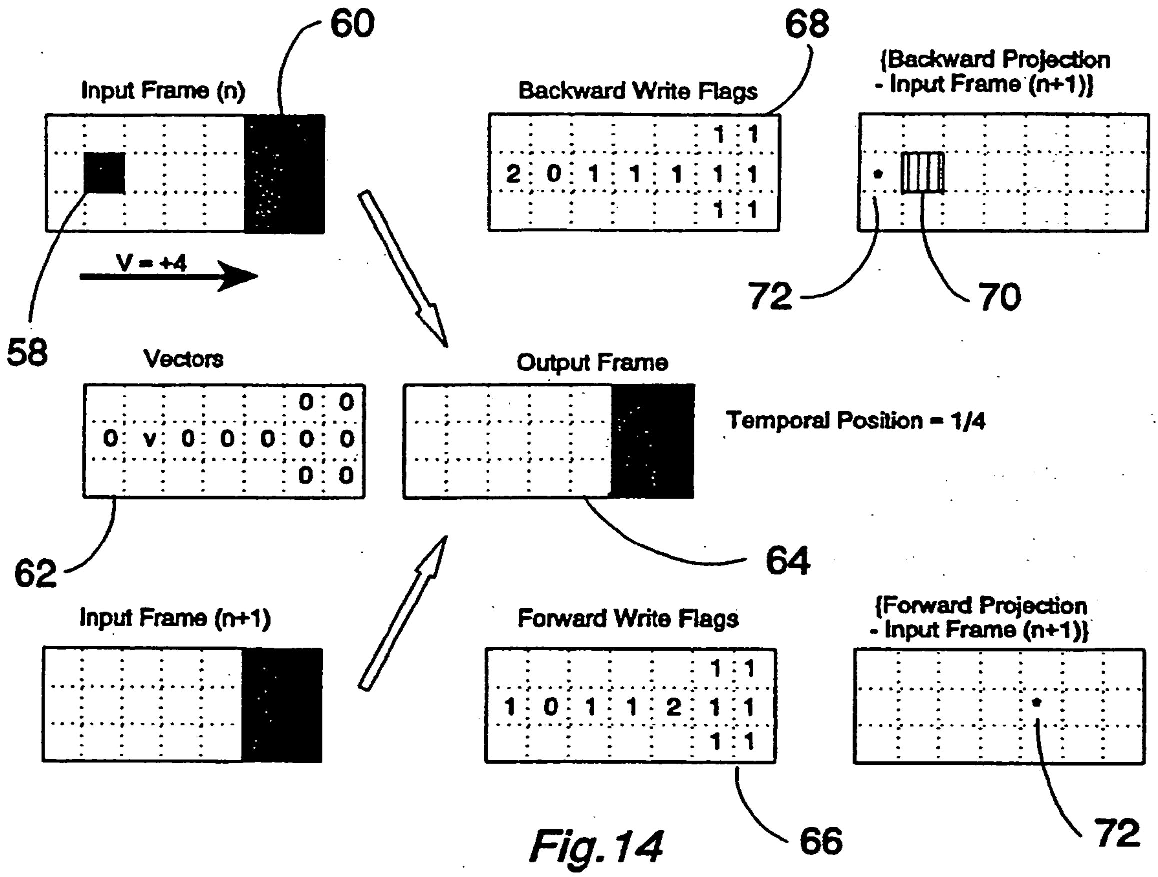

60
68
Input Frame (n)
Backward Write Flags
{Backward Projection - Input Frame (n+1)}
1 1
2 0 1 1 1 1 1
1 1
V = +4
72
70
58
Vectors
Output Frame
0 0
0 v 0 0 0 0 0
0 0
Temporal Position = 1/4
62
64
Input Frame (n+1)
Forward Write Flags
{Forward Projection - Input Frame (n+1)}
1 1
1 0 1 1 2 1 1
1 1
66
72

Fig. 14

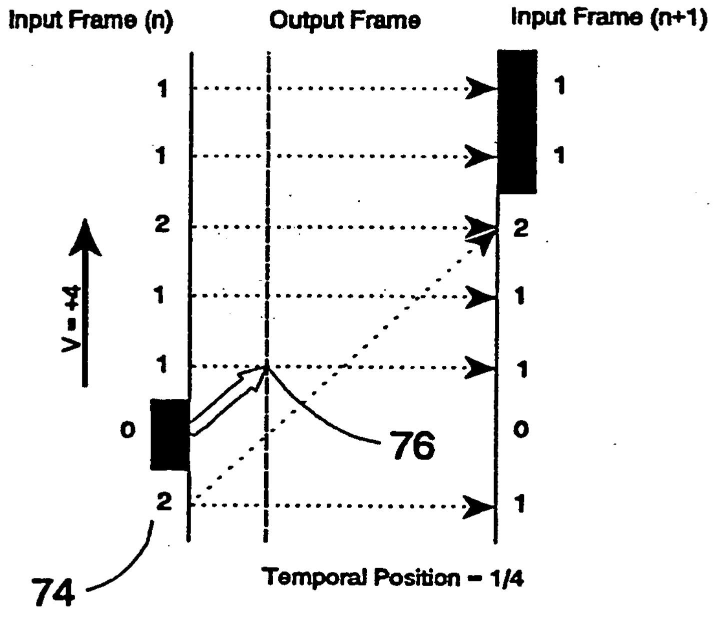

Input Frame (n)
Output Frame
Input Frame (n+1)
1
1
2
1
1
0
2
1
1
2
1
1
0
1
V = +4
76
74
Temporal Position = 1/4

Fig. 15

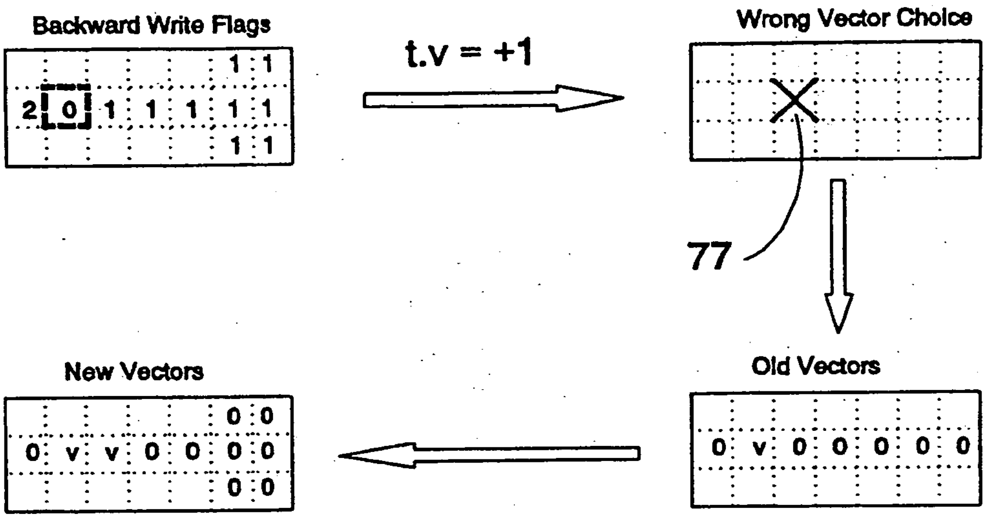

Backward Write Flags
1 1
2 0 1 1 1 1 1
1 1
t.v = +1
Wrong Vector Choice
77
Old Vectors
0 v 0 0 0 0 0
New Vectors
0 0
0 v v 0 0 0 0
0 0

Fig.16

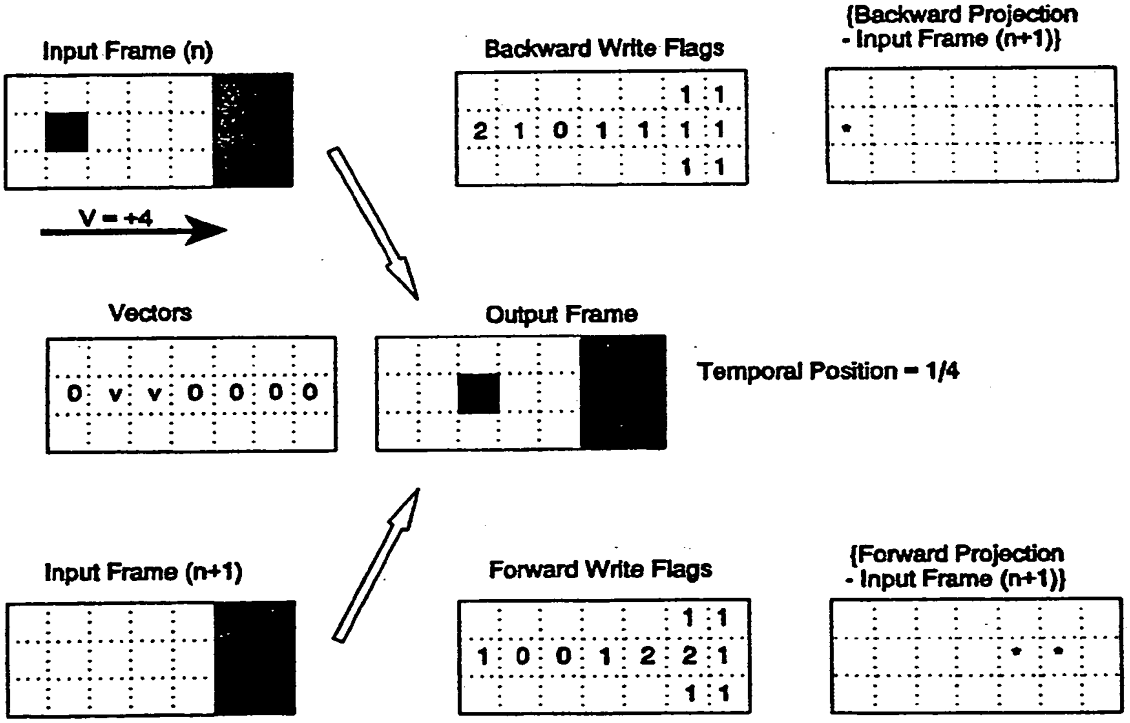

Input Frame (n)
V = +4
Backward Write Flags
1 1
2 1 0 1 1 1 1
1 1
{Backward Projection
- Input Frame (n+1)}
Vectors
0 v v 0 0 0 0
Output Frame
Temporal Position = 1/4
Input Frame (n+1)
Forward Write Flags
1 1
1 0 0 1 2 2 1
1 1
{Forward Projection
- Input Frame (n+1)}

Fig.17

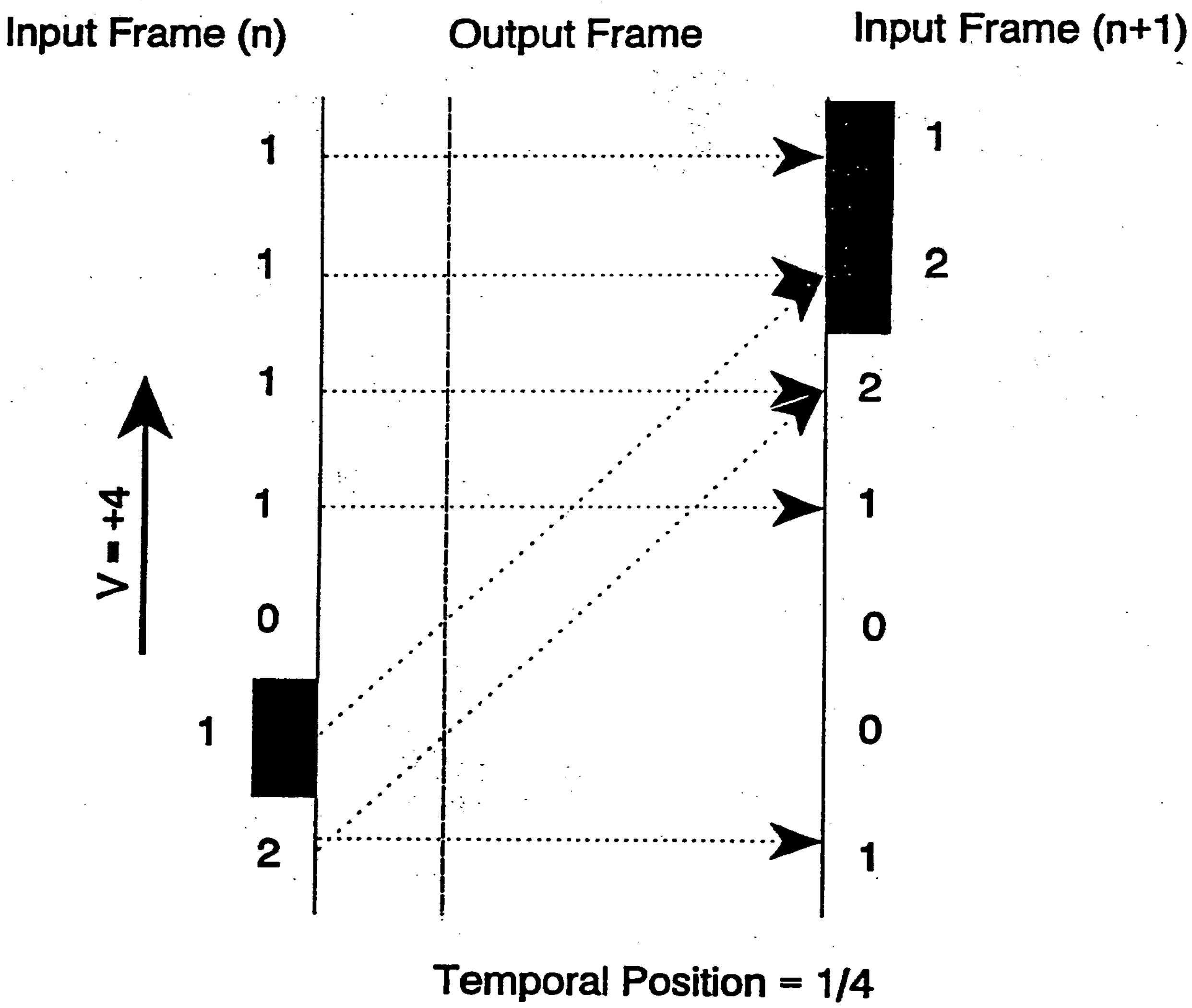

Input Frame (n)
Output Frame
Input Frame (n+1)
1
1
1
1
0
1
2
1
2
2
1
0
0
1
V = +4
Temporal Position = 1/4

Fig.18

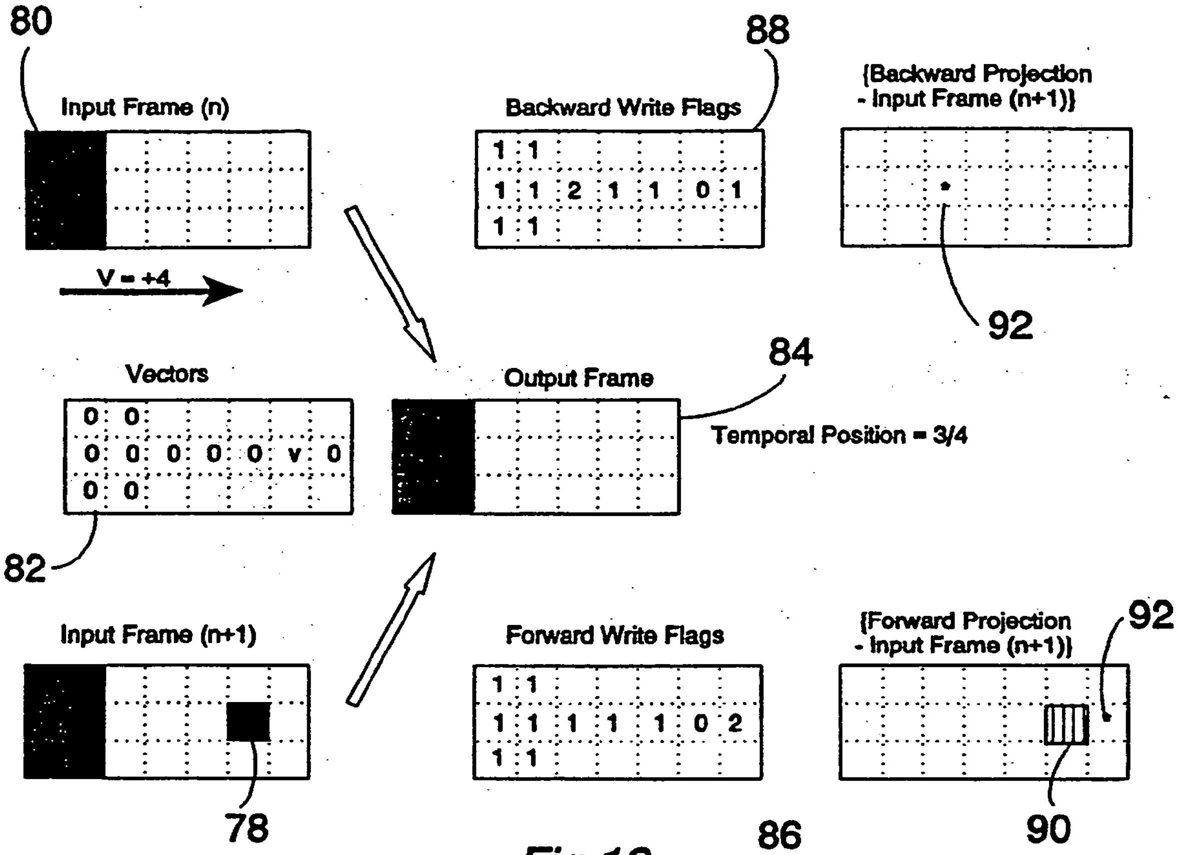

80
Input Frame (n)
V = +4
88
Backward Write Flags
1 1
1 1 2 1 1 0 1
1 1
{Backward Projection
- Input Frame (n+1)}
92
Vectors
0 0
0 0 0 0 0 v 0
0 0
82
Output Frame
84
Temporal Position = 3/4
Input Frame (n+1)
78
Forward Write Flags
1 1
1 1 1 1 1 0 2
1 1
86
{Forward Projection
- Input Frame (n+1)}
92
90

Fig.19

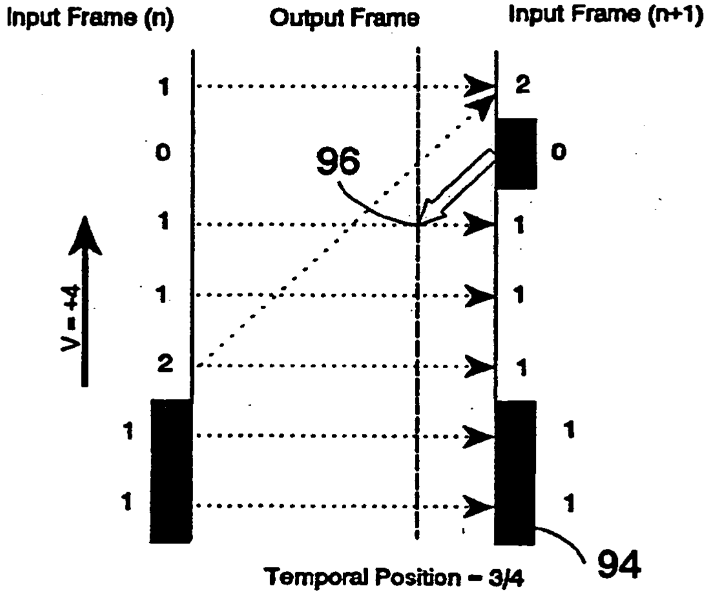

Input Frame (n)
Output Frame
Input Frame (n+1)
1
0
1
1
2
1
1
2
0
1
1
1
1
1
96
V = +4
Temporal Position = 3/4
94

Fig.20

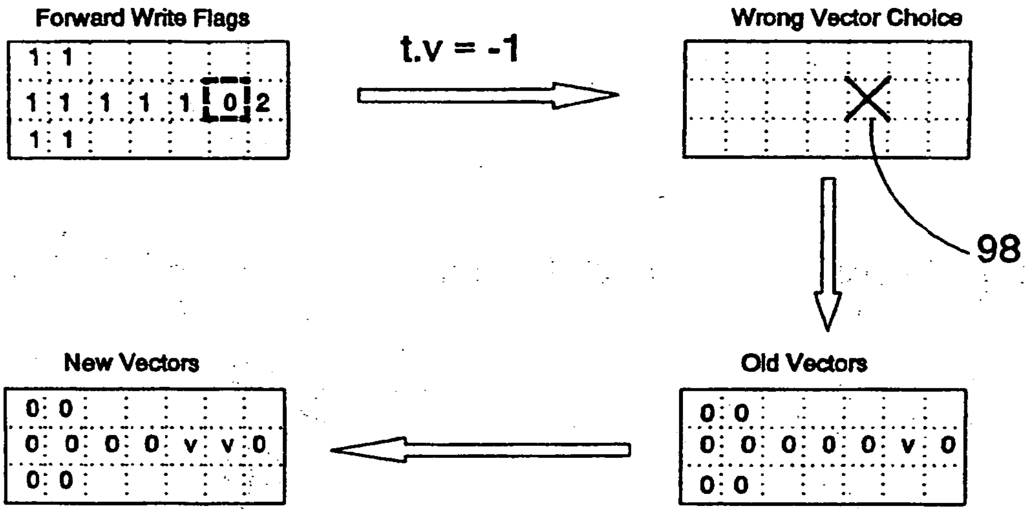

Forward Write Flags
1 1
1 1 1 1 1 0 2
1 1
t.v = -1
Wrong Vector Choice
98
Old Vectors
0 0
0 0 0 0 0 v 0
0 0
New Vectors
0 0
0 0 0 0 v v 0
0 0

Fig.21

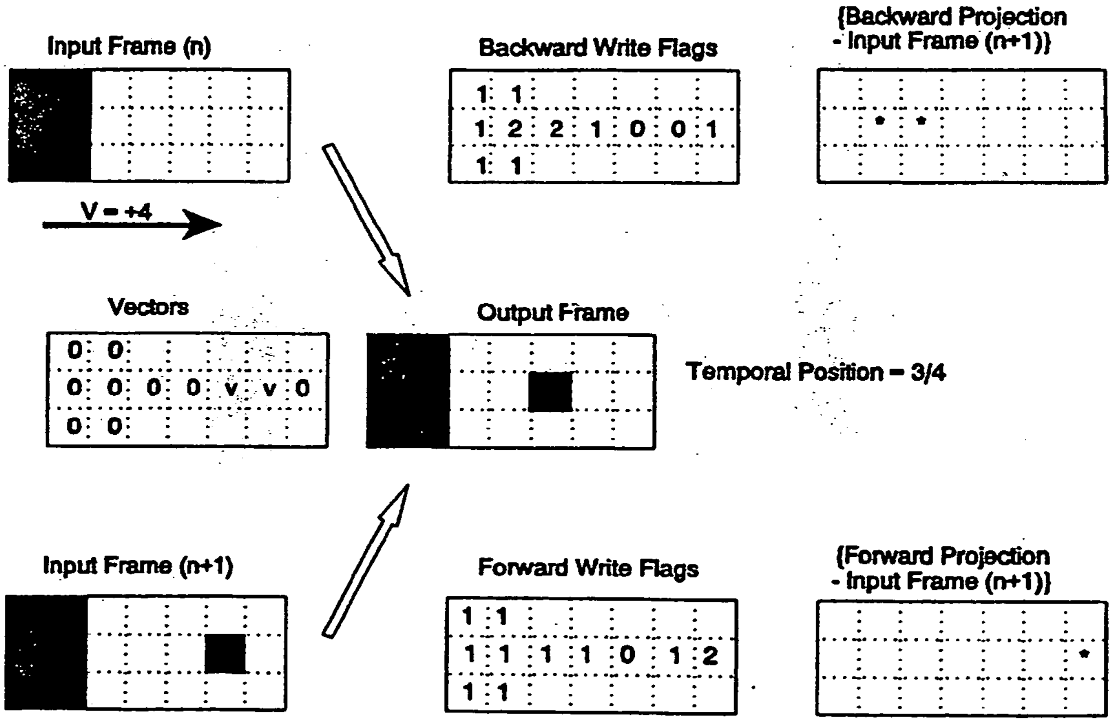

Input Frame (n)
V = +4
Backward Write Flags
1 1
1 2 2 1 0 0 1
1 1
{Backward Projection
- Input Frame (n+1)}
Vectors
0 0
0 0 0 0 v v 0
0 0
Output Frame
Temporal Position = 3/4
Input Frame (n+1)
Forward Write Flags
1 1
1 1 1 1 0 1 2
1 1
{Forward Projection
- Input Frame (n+1)}

Fig.22

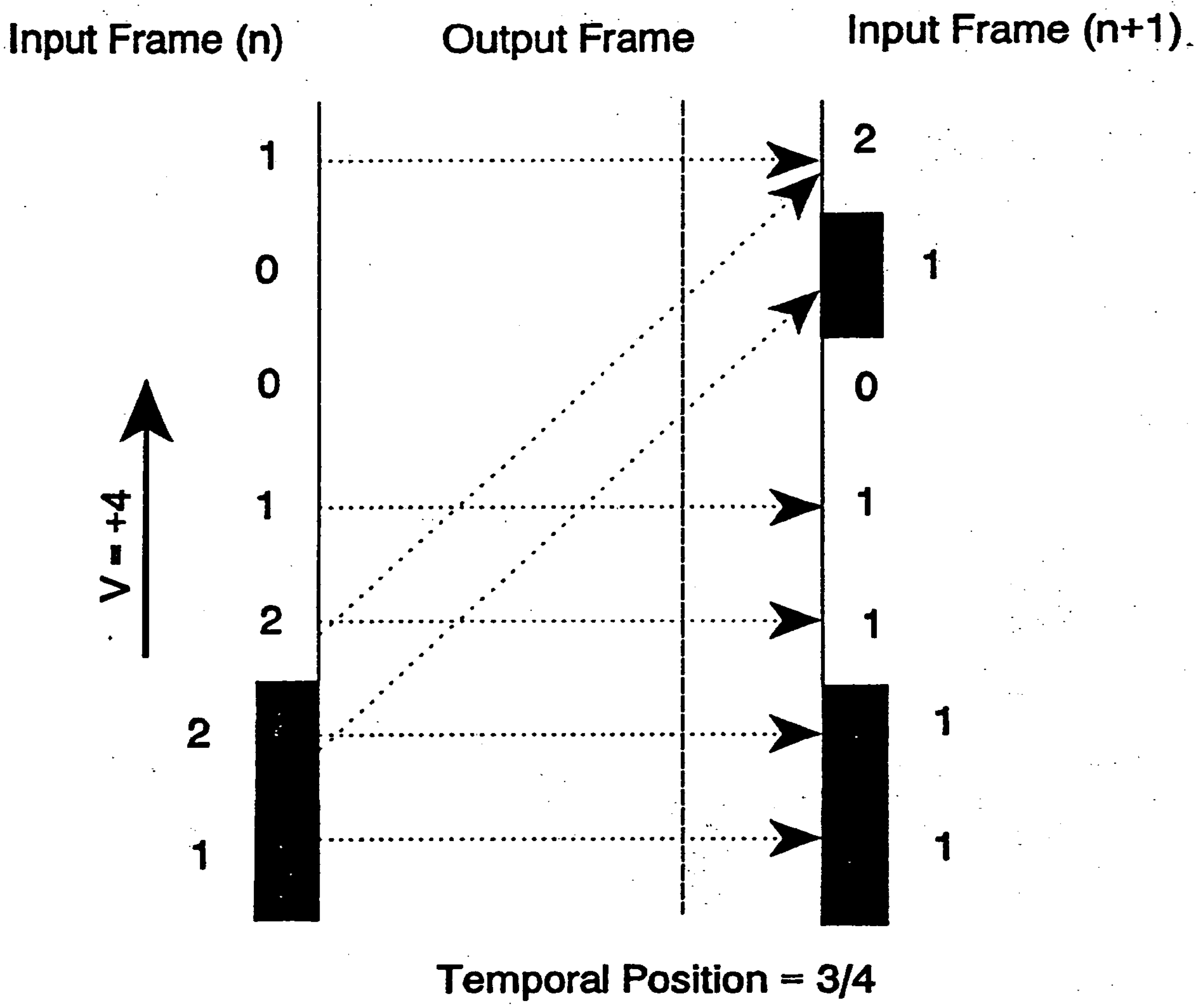

Input Frame (n)
Output Frame
Input Frame (n+1)
1
0
0
1
2
2
1
V = +4
2
1
0
1
1
1
1
Temporal Position = 3/4

Fig.23

MOTION COMPENSATED IMAGE INTERPOLATION

This invention relates to the field of motion compensated image interpolation.

Motion compensated image interpolation for purposes such as standards conversion is known, e.g. film to television, or from one television format to another. An example of such a system is described in British Published Patent Application GB-A-2 231 749 (Sony Corporation). A problem that occurs in these systems when there is relative motion between objects in the foreground and objects in the background is that parts of the objects in the background may be covered or uncovered by the objects in the foreground between successive input images.

An example of covering and uncovering of background objects is shown in Figure 1 of the accompanying drawings. As the car in the foreground passes a stationary background, it uncovers the front wheel of the tow truck and covers the base of the radio mast. Covering and uncovering must be taken into account in the interpolation of an output field between the two fields shown in Figure 1. In particular, in the output field, the radio mast and the tow truck wheel may be partially covered/uncovered, depending upon the temporal position of the output field. This has the result that only pixels from field 0 are available to generate the visible part of the radio mast, and only pixels from field 1 are available to generate the visible part of the front wheel of the tow truck.

It is therefore necessary to detect the covering or uncovering of an object, in order that the interpolator can be steered only to use pixels from the appropriate one of the input fields (i.e. field 0 for the radio mast and field 1 for the tow truck wheel in the above example). If this is not achieved or some other corrective measure taken, then a degradation in quality of the interpolated images will result.

Viewed from one aspect the invention provides an apparatus for performing motion compensated image interpolation between temporally adjacent input arrays of input pixel values to generate an output array of output pixel values, said apparatus comprising:

means for detecting a primary array of motion vectors associated with output pixel positions and representing image motion at said output pixel positions between said input arrays of input pixel values;

means for projecting said motion vectors from said output pixel positions to said input arrays of input pixel values to detect how many times each input pixel value is used as a source for an output pixel value; and

means for controlling subsequent operation to generate said output array of output pixel values in response to said detection of how many times each input pixel value is used as a source for an output pixel value.

Projection of the motion vectors on to the input arrays of input pixel values to determine how many times each input pixel value is used as a source for interpolation serves to detect areas of image that are being either covered or uncovered. With this information, subsequent operation (e.g. interpolator steering, motion vector testing and substitution, etc.) can be controlled in a variety of different ways so as to improve the interpolated image quality despite the regions of cover and uncover.

Whilst the means for controlling subsequent interpolation can take several different forms, in preferred embodiments said means for controlling comprises:

means for interpolating a test output array of test output pixel values using said primary array of motion vectors;

means for comparing said test output array of test output pixel values with said input arrays of input pixel values to identify drop out error positions at which corresponding pixel values at positions compensated for motion with said primary array of motion vectors projected from said input arrays of pixel values differ between said test output array of test pixel values and either of said input arrays of input pixel values and at which those input pixel values have both not been used as a source;

means for detecting if primary motion vectors associated with said drop out error positions project to positions in said input arrays of input pixel values at which those input pixel values have both not been used once as a source to identify confirmed drop out error positions;

means for projecting primary motion vectors at said confirmed drop out error positions to said primary array of motion vectors to identify erroneous motion vectors;

means for selecting alternative motion vectors to replace said erroneous motion vectors to produce a secondary array of motion vectors; and

means for interpolating said output array of output pixel values using said secondary array of motion vectors.

With this technique the control of subsequent interpolation takes the form of identifying erroneous motion vectors and replacing them with alternatives so as to produce a better quality interpolated image.

An alternative preferred technique for controlling subsequent interpolation is one in which said means for controlling comprises:

means for detecting pixel source error positions in said input arrays of input pixel values at which input pixel values have been used as a source more than once;

means for flagging output pixel positions coincident with said pixel source error positions;

means for setting a flag for those output pixel positions surrounding each pixel source error position within an error area with dimensions proportional to that motion vector at said pixel source error position and temporal displacement between said output array of output pixel values and that input array of input pixel values from which said flag arose; and

means for interpolating said output array of output pixel values for flagged output pixel positions using one or more input pixel values from that input array of input pixel values from which said flag did not arise.

With this technique, the detection of regions of cover and uncover is used to steer the interpolator to use the appropriate input array of input pixel values to the exclusion of that in which the input pixel values has been used as a source more than once.

As a refinement to this technique to take account of errors in the motion vectors associated with regions of cover and uncover or for use in its own right, in preferred embodiments of the invention said means for controlling comprises:

means for detecting pixel source error positions in said input arrays of input pixel values at which input pixel values have been used as a source more than once;

means for flagging output pixel positions coincident with said pixel source error positions; and

means for interpolating said output array of output pixel values for flagged output pixel positions using motion vectors at a corresponding position with said flagged output pixel positions associated with a temporally adjacent output array of output pixel values in a temporal direction of that input array of input pixel values from which said flag did not arise.

It will be appreciated that detecting how many times each input pixel value has been used as a source is a computationally intensive activity. Whilst a general purpose computer could perform this function, preferred embodiments incorporate special purpose hardware such that said means for projecting comprises:

means for storing an array of marker values indicating how many time each input pixel value is used as a source for an output pixel value;

means for generating a stream of motion vectors and their associated output pixel position addresses;

means for calculating an address offset from each output pixel position to each of said temporally adjacent input arrays of input pixel values by multiplying that motion vector associated with said output pixel position by temporal displacement to said input arrays of input pixel values;

means for adding each of said address offsets to said associated output pixel position address to yield source addresses in each of said temporally adjacent input arrays of input pixel values; and

means for incrementing said marker values for said source addresses to indicate its use as a source.

Viewed from another aspect the invention provides a method of performing motion compensated image interpolation between temporally adjacent input arrays of input pixel values to generate an output array of output pixel values, said method comprising the steps of:

detecting a primary array of motion vectors associated with output pixel positions and representing image motion at said output pixel positions between said input arrays of input pixel values;

projecting said motion vectors from said output pixel positions to said input arrays of input pixel values to detect how many times each input pixel value is used as a source for an output pixel value; and

controlling subsequent operation to generate said output array of output pixel values in response to said detection of how many times each input pixel value is used as a source for an output pixel value.

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 illustrates the problem of the cover and uncover of regions in adjacent images;

Figures 2 and 3 illustrate steered interpolation using hitboards (i.e. arrays indicating how many times each pixel has been used as a source for interpolation);

Figure 4 illustrates a circuit for generating hitboard data;

Figure 5 illustrates a circuit for modified interpolation on the basis of hitboard data;

Figures 6, 7 and 8 illustrate motion vector selection and steered interpolation on the basis of hitboard data; and

Figures 9 to 13, 14 to 18 and 19 to 23 illustrate the detection and replacement of erroneous motion vectors utilising hitboard data.

Figure 1 illustrates two temporally adjacent fields that form input arrays of input image values between which it is desired to interpolate an output frame of output image values. The tow truck and radio mast are stationary in both field 0 and field 1. The car moves from in front of the tow truck in field 0 to in front of the radio mast in field 1. Thus, in field 1 the lower cab and front wheel of the tow truck has been uncovered whilst the lower portion of the radio mast has been covered.

Depending upon the relative temporal position of the output frame to be interpolated, it may be necessary that the output frame contain a region of image that is either uncovered or covered in field 1. In order to be accurately interpolated, the uncovered region in the interpolated field should be derived solely from field 1 and conversely, the covered region should be interpolated entirely from field 0. In order to enable such steered interpolation yielding more accurate interpolation, regions that are covered and uncovered between temporally adjacent frames must be detected.

Figure 2 illustrates the use of a hitboard technique for detecting regions of cover and uncover. Field 0 and field 1 show a line of letters moving leftwards (i.e. v = -3) behind a stationary (i.e. v = 0) block 2. The output field to be interpolated is at a position two thirds of the way towards field 1 (i.e. t = 2/3).

In field 0 the rightmost letter visible is D. In field 1 the rightmost letter visible is G. In the interpolated field the rightmost letter that should be seen is F. It is assumed that the correct motion vectors have been established for all of the pixels in the output field (i.e. v = -3 for letters A to F and v = 0 for all of the pixels of the block 2).

Under known operation with previous devices, each output pixel is derived by projecting its motion vector forwards and backwards to the temporally adjacent fields and using a sum of the pixel values in the input images pointed to, each being weighted according to the relative temporal proximity of the output field to its input fields. In the present case, the letters E and F do not appear in field 0 and so any contribution from field 0 would be likely to degrade the image quality.

The hitboard for field 0, hitboard 0, is derived by projecting backwards from the output field positions using each of the motion vectors and incrementing a stored value for the associated pixel in the field 0. All of the input pixel values shown in field 0 are used as a source once apart from the two input field values at the left hand edge of the block 2.

Multiple hitboard entries in the temporally backward direction indicate a region of uncover. The control flags associated with the output field for the two pixels in which the double hits occur are set to "1" to indicate that interpolation should occur wholly from field 1. In addition, the pixels on either side of those for which the control flags have been set are examined to see what motion vector is associated with them. The control flags are then extended to either side by a distance given by the motion vector for the adjacent pixel multiplied by the temporal displacement from the field in which the double hit occurred.

The remaining control flags where single hits occurred are set to "b" to indicate that both field 0 and field 1 should be used as sources for interpolation.

horizontally and $-y_bt$ and $+y_tt$ pixels vertically of the set control flag will also be set.

Figure 3 illustrates another example of the above technique. In this example, the line of letters is moving leftwards at v = -6 and the block 2 is moving rightwards at v = +3. Backwards projection from the output field results in the letters E to J projecting onto the leftmost six pixels of the block 2 in field 0 in addition to the true projection from the leftmost six pixels of the block 2 in the output field. Accordingly, these six pixels have multiple hitboard entries indicating that interpolation for them should occur only from field 1. In addition to the setting of those control flags directly corresponding to the multiple hitboard entries, the motion vectors associated with adjacent pixels p, q are assessed to be -6 and +3 respectively. Multiplying each of these by the temporal displacement (t=2/3) from the field 0 in which the multiple entry occurred extends the control flag "1" by four pixel positions leftwards and two pixel positions rightwards.

In this way, the pixels E to J are correctly interpolated from only field 1 thereby improving interpolated image quality.

Figure 4 illustrates a circuit for deriving the entries to be made in the hitboards for temporally adjacent forwards and backwards fields/frames. A pixel address counter 8 generates a sequence of pixel addresses moving through the output field to be interpolated. In synchronism with the pixel address counter 8, the motion vectors at the pixel addresses are supplied through input 10 to two multipliers 12, 14. The output field to be interpolated is at a temporal position t between the temporally backward field 0 and the temporally forward field 1. The temporal displacement from the output field to the field 0 of "-t" is also applied to the multiplier 12 and a corresponding value "(1-t)" is applied to the other multiplier 14 to represent the temporal displacement from the output field to the field 1.

The output from the multiplier 12 represents the offset to the pixel address that is produced by projecting the vector at the pixel position corresponding to the pixel address currently being output from the pixel address counter 8 backwards onto field 0. Similarly, the output from the multiplier 14 represents the corresponding offset to

With the control flags thus set, the output image pixel values for the letters E and F are derived wholly from field 1 with no contribution from field 0. This improves interpolated image quality.

As can be seen from the above, the hitboards are employed to steer the interpolator, so that covered or uncovered parts of the image are derived from only the appropriate one of the input fields. The technique assumes that the correct motion vectors have been selected for each output pixel.

An array of control flags (one per pixel of the output field) is set up, to indicate one of the following options;

(a) interpolated from field 0 only;

(b) interpolate from field 1 only;

(c) interpolate from both output fields.

The flags are generated as follows;

1. Examine the two hitboards for double (or multiple) entries. If a more than once entry is found in hitboard 0, then only field 1 should be used for generating the output pixel at that position. Similarly, if a more than once entry is found in hitboard 1, then only field 0 should be used to generate the output pixel at that position. The control flags for those positions are set accordingly.

2. Where a control flag is set to indicate that only one field should be used in interpolation of that output pixel, a number of adjacent control flag are set. At each edge of an area of set control flags, set the n adjacent control flags to the same setting. Here, n is equal to the vector component at that edge in the direction of the adjacent flag (i.e. horizontally, vertically, up or down), multiplied by the temporal position (t) of the output field. Thus, if the horizontal vector component at the left edge is x_l, the horizontal vector component at the right edge is x_r, the vertical vector component at the top edge is y_t, the vertical vector component at the bottom is y_b and the temporal position is t (where $0 \le t \le 1$), then the control flags that are within $-x_l t$ and $+x_r t$ pixels

the pixel address for the projection forwards onto field 1. These address offsets are supplied to respective adders 16 and 18 where they are summed with the current pixel address from the pixel address counter 18 to yield source addresses for projected interpolation in the backward and forward fields. These source addresses are applied to the hitboards 20 and 22 (arrays of RAM with a storage location for each pixel position) where they cause the value corresponding the address pointed to be incremented to indicate its intended use as a source of interpolation.

Figure 5 illustrates a circuit for controlling interpolation utilising hitboards. Vectors generated in the known manner are supplied to a vector selector 24 from which they pass to a vector buffer 26 and a hitboard generator 28. The hitboard generator 28 has the form described in relation to Figure 4. The vector buffer 26 serves to delay the vectors for a period corresponding to the processing lag in the upper arm of the circuit of Figure 5. The hitboards 20 and 22 for respective backwards and forwards projections from the output field in conjunction with the multiple entry detector 30 serve to detect areas of cover and uncover in the image. The multiple entry detector 30 supplies signals indicative of multiple entries to a control flag generator 32 that serves to produce the control flags that steer the interpolator and also performs the edge extension of the control flag discussed above. The control flag generator 32 is supplied with the delayed vectors from the vector buffer and the temporal position of the output field in addition to the output from the multiple entry detector 30 as this information is needed to generate the control flags to the interpolator (downstream) that control whether field 0, field 1 or both fields are used as the source of interpolation.

Figure 6 illustrates the use of hitboard entries to control subsequent interpolation by forcing the use of motion vectors from the previous or next output fields in the interpolation of covered and uncovered objects respectively. The sequence of input frames i/p1, i/p2 and i/p3 show a block 34 moving rightwards against a background 36 that is moving leftwards. Output frames o/p1, o/p2 are to be interpolated at positions between the input frames. The pixel indicated by an "o" are uncovered by the divergence between the respective scenes and do not have valid motion vectors associated with them.

The default motion vector associated with the uncovered pixels a is zero. This leads to multiple entries in the backward hitboard to input field i/p1 at the points indicated by an "x". The other uncovered pixels are at positions where the motion vectors are derived from the moving block and at which no multiple entries in the backwards hitboard occur.

The multiple entries in the backwards hitboard at the positions "x" indicate divergence and that the motion vectors should be taken from the succeeding frame pair i/p2 and i/p3. These motion vectors 38 can then be used in the interpolation of the points 40 for which motion vectors had not previously been correctly identified. The interpolation may then proceed using these corrected motion vectors according to the technique illustrated in Figures 2 and 3.

In a similar manner, the pixels b are uncovered between input frames i/p2 and i/p3 and yield multiple entries in the backwards hitboard. Accordingly, the motion vectors associated with the succeeding input frames are used for the pixels 42 interpolated in the output field o/p2.

Figure 7 illustrates the situation for convergence. In this case, the block 34 and the background 36 are converging at the right hand edge of the block 34. The pixel positions c in the output frame o/p1 are hidden by the convergence with the result that they have a default zero motion vector that yields a double hitboard entry in the forward direction corresponding to input field i/p2. The forward direction indicates that the vectors for the points should be taken from the backwards direction, i.e. from vectors derived between input fields i/p0 (not shown) and i/p1. As before, these corrected motion vectors can then be used for interpolation.

In a similar manner to the above, the motion vectors for the pixels d can be taken from those between input frames i/p1 and i/p2 in view of the double hitboard entry in the forward direction.

The above operation can be considered as follows:

1. Converging vectors (i.e. multiple writes) in the backward direction indicate a diverging scene, hence vectors will have to be obtained from the next choice of output vectors.

OR diverging vectors (i.e. zero writes) in the forward direction indicate a diverging scene, hence vectors will have to be obtained from the next choice of output vectors.

2. Diverging vectors (i.e. zero writes) in the backward direction indicate a converging scene, hence vectors will have to be obtained from the previous choice of output vectors.

OR converging vectors (i.e. multiple writes) in the forward direction indicate a converging scene, hence vectors will have to be obtained from the previous choice of output vectors.

Figure 8 illustrates a sequence of input fields i/p1 to i/p6. In this sequence a dark block starts and then stops moving against a background whilst a camera tries to track it. Between frames i/p2 and i/p3 the block moves and whilst the camera tries to follow it (indicated by moving background), the block still moves in the frame. This is also the case between frames i/p3 and i/p4. Between frames i/p4 and i/p5 the camera is successfully following the block holding it stationary within the frame whilst the background moves. Between frames i/p5 and i/p6 the block stops moving. Output fields o/p1, o/p2 and o/p3 are at positions between the input fields.

In this example, it will be seen that when a poor match occurs in vector selection coincident with zero writes in the backward write projection, then the vectors used should be those used in the previous output field forward projected.

In Figure 8, it can be seen that the vectors chosen in this way are a, b, c, d, i and j. The vectors a, b, c and d are in fact the wrong vectors, but represent the best "available" vector in the circumstances. The vectors i and j are the correct vectors.

Conversely, when a poor match occurs in vector selection coincident with zero writes in the forward write projection then the vectors used should be those used in the next output field backward projected. In this case these vectors are k, l, m, n, o and p, which in each case are the correct vectors. This is also the case for vectors e, f, g and h.

Figures 9 to 13 illustrate a third technique in which hitboard entries are used to modify subsequent interpolation so as to improve output image quality. In this case, the hitboards are used, with other information, to correct erroneously selected motion vectors.

The processing takes place as part of the interpolation of the output field. A test interpolation is first performed, and the motion vectors and interpolator steering modified (as necessary) for final interpolation. Hitboards (backwards/forwards write flags) are set up indicating the intersection of the selected motion vector for each output pixel with the two input fields. In addition, the output frame generated in the test interpolation is projected onto the input frames that generated it by running all the motion vectors backwards (i.e. a sort of reverse interpolation). If the test interpolation was accurate, then this projection should regenerate the input frames (this comparison will be referred to as forward and backward projection). The reverse projected frames are subtracted, on a pixel by pixel basis, from the true input frames to produce error frames. If the interpolation was completely accurate, then all the pixels in the error frames would be zero. Positions of inaccurate interpolation are indicated by non-zero values in the error frames.

As shown in Figures 9 and 10 a pixel 38 moves horizontally by four pixels between input frame (n) and input frame (n+1). The motion vectors selected by conventional means are shown in the array 40. The two input frames and the motion vectors are then used to generate a test output frame 42 and to generate the forward and backward hitboards 44 and 46 respectively. It will be seen that for the middle pixel three positions from the left in Figure 9, a match of a white pixel to a white pixel with zero motion is just as good an apparent match as the true black to black pixel match for the moving pixel 38.

Forward and backward projection is then performed by subtracting each output pixel in the reverse projection from the corresponding position in input frames (n) and (n+1) to generate the forward and backward error frames. Differences between the input pixels and the forward and backward projected pixels are marked 48, 50 as error positions in the input frame coordinates (the differences are compared with a threshold value, differences exceeding the threshold value being marked as error positions, filtering would also be used to reduce the effect of objects and object edges that fall partially in one pixel and partially in it neighbour). Pixels for which two or more entries are present in the hitboards are ignored (masked) in this error marking as is indicated by the "*" 52.

The motion vectors are then corrected by the following set of rules:

1. Examine the hitboards, looking for a pixel position having zero entries in the forward and backward hitboards.
2. Does a discrepancy in the forward or backward error frames occur at this point (i.e. an error position)?
3. Does the vector used at this position point to a multiple hitboard entry?
4. If the answers to (2) and (3) are yes, then a vector can be identified that is probably erroneous. That vector can be discarded and a neighbouring (different) vector tried. The position of the erroneous vector is derived as follows:

    if the present vector is positive and t< 0.5, then the position of the erroneous vector is displaced by +(t.v) with respect to the zero write flag in the backward hitboard;

    if the present vector is positive and t > 0.5, then the position of the erroneous vector is displaced by -((1-t).v) with respect to the zero write flag in the forward hitboard.

Returning now to Figure 10, this illustrates the test interpolation and hitboard generation for the image regions shown in Figure 9. In particular the numbers of times each pixel is used as a source is recorded in the hitboards as indicated by the numbers 54. It will be seen that the pixel 38 is not used as a source in either the forward or reverse input frames and thus the interpolated output frame does not contain this dark pixel 38. Accordingly, an error position will be detected in the backward and forward error frames at the positions corresponding to the dark pixel 38 in the input frames n and (n+1).

As shown in Figure 11, the error position 48 from the backward error frame in Figure 9 is searched in the hitboards 44 and 46 to see if it corresponds to a position not used as a source. In this case it does so correspond and accordingly an erroneous vector is probably present. Since this error is detected in the backward error frame, there is no need to correct for the same error in the forward projection. The position of this vector is calculated as t.v from the position in the backward direction, i.e. in this case one pixel displacement to the right at position 56. The old vector at this position (a zero motion vector) is replaced with an alternative vector chosen from nearby in accordance with usual vector selection practice (in this case v).

Figures 12 and 13 illustrate what would occur when this test is performed with the corrected motion vectors, i.e. a situation in which no errors are present. As will be seen, the pixel 38 in this case appears in the output frame. Further, the backward and forward error frames do not contain any error positions, i.e. pixels that do not project from their input frame to a corresponding pixel in the output frame.

Figures 14 to 18 illustrate another example of the third technique in which a dark pixel 58 moves behind a stationary dark area 60, i.e. convergence. With reference to Figure 14, the pixel 58 moves horizontally by four pixels between input frame (n) and input frame (n+1). The motion vectors selected by conventional means are shown in the array 62. The two input frames and the motion vectors are then used to generate a test output frame 64 and to generate the forward and backward hitboards 66 and 68 respectively.

Forwards and backwards projection is performed and then these frames are compared with the corresponding input frames to generate forward and backward error frames. Differences between the input pixels and the forward and backward projected pixels are marked 70 as error positions in the input frame coordinates (the differences are compared with a threshold value, differences exceeding the threshold value being marked as error positions; filtering would also be used to reduce the effect of objects and object edges that fall partially in one pixel and partially in it neighbour). Pixels for which two or more entries are present in the hitboards are ignored (masked) in this error marking as is indicated by the "*" 72.

The motion vectors are then corrected by the set of rules as given above.

Figure 15, this illustrates the test interpolation and hitboard generation for the image regions shown in Figure 14. In particular the numbers of times each pixel is used as a source is recorded in the hitboards as indicated by the numbers 74. It will be seen that the pixel 58 is not used as a source in either the forward or reverse input frames and thus the interpolated output frame does not contain this dark pixel 58. Accordingly, an error position is detected in the backward error frame at the position corresponding to the dark pixel 58 in the input frame n.

As shown in Figure 16, the error position 70 from the backward error frame of Figure 14 is searched in the hitboards 66 and 68 to see if it corresponds to a position not used as a source. In this case it does so correspond and accordingly an erroneous vector is probably present. The position of this vector is calculated as t.v from the position in the backward direction, i.e. in this case one pixel displacement to the right at position 77. The old vector at this position (a zero motion vector) is replaced with an alternative vector chosen from nearby in accordance with usual vector selection practice (in this case v).

Figures 17 and 18 illustrate what would occur when this test is performed with the corrected motion vectors, i.e. a situation in which no errors are present. As will be seen, the pixel 58 in this case appears in the output frame. Further, the backward and forward error frames do not contain any error positions, i.e. pixels that do not project from their input frame to a corresponding pixel in the output frame.

Figures 19 to 23 illustrate another example of the third technique in which a dark pixel 78 moves out from behind a dark area 80 (divergence). With reference to Figure 19, the dark pixel 78 moves horizontally by four pixels between input frame (n) and input frame (n+1). The motion vectors selected by conventional means are shown in the array 82. The two input frames and the motion vectors are then used to generate a test output frame 84 and to generate the forward and backward hitboards 86 and 88 respectively.

Forward and backward projection is then performed by subtracting each output pixel in the reverse projection from the corresponding position in input frames (n) and (n+1) to generate the forward and backward error frames. Differences between the input pixels and the forward and backward projected pixels are marked 90 as error positions in the input frame coordinates (the differences are compared with a threshold value, differences exceeding the threshold value being marked as error positions; filtering would also be used to reduce the effect of objects and object edges that fall partially in one pixel and partially in it neighbour). Pixels for which two or more entries are present in the hitboards are ignored (masked) in this error marking as is indicated by the "*" 92.

The motion vectors are then corrected by the set of rules given above.

Figure 20 illustrates the test interpolation and hitboard generation for the image regions shown in Figure 19. In particular the numbers of times each pixel is used as a source is recorded in the hitboards as indicated by the numbers 94. It will be seen that the pixel 78 is not used as a source in either the forward or reverse input frames and thus the interpolated output frame does not contain this dark pixel 78. Accordingly, an error position will be detected in the forward error frame at the position corresponding to the dark pixel 78 in the input frame (n+1).

As shown in Figure 21, the error position 90 from the forward error frame in Figure 19 is searched in the hitboards 86 and 88 to see if it corresponds to a position not used as a source. In this case it does so correspond and accordingly an erroneous vector is probably present. Since this error is detected in the forward error frame 86, there is no need to correct for the same error in the backward error frame. The position of this vector is calculated as -(1-t).v from the position in the forward direction, i.e. in this case one pixel displacement to the left at position 98. The old vector at this position (a zero motion vector) is replaced with an alternative vector chosen from nearby in accordance with usual vector selection practice (in this case v).

Figures 22 and 23 illustrate what would occur when this test is performed with the corrected motion vectors, i.e. a situation in which no errors are present. As will be seen, the pixel 78 in this case appears in the output frame. Further, the backward and forward error frames do not contain any error positions, i.e. pixels that do not project from their input frame to a corresponding pixel in the output frame.

CLAIMS

1. Apparatus for performing motion compensated image interpolation between temporally adjacent input arrays of input pixel values to generate an output array of output pixel values, said apparatus comprising:

means for detecting a primary array of motion vectors associated with output pixel positions and representing image motion at said output pixel positions between said input arrays of input pixel values;

means for projecting said motion vectors from said output pixel positions to said input arrays of input pixel values to detect how many times each input pixel value is used as a source for an output pixel value; and

means for controlling subsequent operation to generate said output array of output pixel values in response to said detection of how many times each input pixel value is used as a source for an output pixel value.

2. Apparatus as claimed in claim 1, wherein said means for controlling comprises:

means for interpolating a test output array of test output pixel values using said primary array of motion vectors;

means for comparing said test output array of test output pixel values with said input arrays of input pixel values to identify drop out error positions at which corresponding pixel values at positions compensated for motion with said primary array of motion vectors projected from said input arrays of input pixel values differ between said test output array of test pixel values and either of said input arrays of input pixel values and at which those input pixel values have both not been used as a source;

means for detecting if primary motion vectors associated with said drop out error positions project to positions in said input arrays of input pixel values at which those input pixel values have both not been used once as a source to identify confirmed drop out error positions;

means for projecting primary motion vectors at said confirmed drop out error positions to said primary array of motion vectors to identify erroneous motion vectors;

means for selecting alternative motion vectors to replace said erroneous motion vectors to produce a secondary array of motion vectors; and

means for interpolating said output array of output pixel values using said secondary array of motion vectors.

3. Apparatus as claimed in claim 1, wherein said means for controlling comprises:

means for detecting pixel source error positions in said input arrays of input pixel values at which input pixel values have been used as a source more than once;

means for flagging output pixel positions coincident with said pixel source error positions;

means for setting a flag for those output pixel positions surrounding each pixel source error position within an error area with dimensions proportional to that motion vector at said pixel source error position and temporal displacement between said output array of output pixel values and that input array of input pixel values from which said flag arose; and

means for interpolating said output array of output pixel values for flagged output pixel positions using one or more input pixel values from that input array of input pixel values from which said flag did not arise.

4. Apparatus as claimed in claim 1, wherein said means for controlling comprises:

means for detecting pixel source error positions in said input arrays of input pixel values at which input pixel values have been used as a source more than once;

means for flagging output pixel positions coincident with said pixel source error positions; and

means for interpolating said output array of output pixel values for flagged output pixel positions using motion vectors at a corresponding position to said flagged output pixel positions associated with a temporally adjacent output array of output pixel values in a temporal direction of that input array of input pixel values from which said flag did not arise.

5. Apparatus as claimed in any one of the preceding claims, wherein said means for projecting comprises:

means for storing an array of marker values indicating how many time each input pixel value is used as a source for an output pixel value;

means for generating a stream of motion vectors and their associated output pixel position addresses;

means for calculating an address offset from each output pixel position to each of said temporally adjacent input arrays of input pixel values by multiplying that motion vector associated with said output pixel position by temporal displacement to said input arrays of input pixel values;

means for adding each of said address offsets to said associated output pixel position address to yield source addresses in each of said temporally adjacent input arrays of input pixel values; and

means for incrementing said marker values for said source addresses to indicate its use as a source.

6. A method of performing motion compensated image interpolation between temporally adjacent input arrays of input pixel values to generate an output array of output pixel values, said method comprising the steps of:

detecting a primary array of motion vectors associated with output pixel positions and representing image motion at said output pixel positions between said input arrays of input pixel values;

projecting said motion vectors from said output pixel positions to said input arrays of input pixel values to detect how many times each input pixel value is used as a source for an output pixel value; and

controlling subsequent operation to generate said output array of output pixel values in response to said detection of how many times each input pixel value is used as a source for an output pixel value.

7. A method as claimed in claim 6, wherein said step of controlling comprises the steps of:

interpolating a test output array of test output pixel values using said primary array of motion vectors;

comparing said test output array of test output pixel values with said input arrays of input pixel values to identify drop out error positions at which coincident pixel values differ between said test output array of test pixel values and either of said input arrays of input pixel values and at which those input pixel values have both not been used as a source;

detecting if primary motion vectors associated with said drop out error positions project to positions in said input arrays of input pixel values at which those input pixel values have both not been used once as a source to identify confirmed drop out error positions;

projecting primary motion vectors at said confirmed drop out error positions to said primary array of motion vectors to identify erroneous motion vectors;

selecting alternative motion vectors to replace said erroneous motion vectors to produce a secondary array of motion vectors; and

interpolating said output array of output pixel values using said secondary array of motion vectors.

8. A method as claimed in claim 6, wherein said step of controlling comprises the steps of:

detecting pixel source error positions in said input arrays of input pixel values at which input pixel values have been used as a source more than once;

flagging output pixel positions coincident with said pixel source error positions;

setting a flag for those output pixel positions surrounding each pixel source error position within an error area with dimensions proportional to that motion vector at said pixel source error position and temporal displacement between said output array of output pixel values and that input array of input pixel values from which said flag arose; and

interpolating said output array of output pixel values for flagged output pixel positions using one or more input pixel values from that input array of input pixel values from which said flag did not arise.

9. A method as claimed in claim 6, wherein said step of controlling comprises the steps of:

detecting pixel source error positions in said input arrays of input pixel values at which input pixel values have been used as a source more than once;

flagging output pixel positions coincident with said pixel source error positions; and

interpolating said output array of output pixel values for flagged output pixel positions using motion vectors at a coincident position with said flagged output pixel positions associated with a temporally adjacent output array of output pixel values in a temporal direction of that input array of input pixel values from which said flag did not arise interpolating uses motion vectors at a corresponding position with said flagged output pixel positions associated with a temporally adjacent output array of output pixel values in a temporal direction of that input array of input pixel values from which said flag did not arise.

10. A method as claimed in any one of claims 6 to 9, wherein said step of projecting comprises the steps of:

storing an array of marker values indicating how many time each input pixel value is used as a source for an output pixel value;

generating a stream of motion vectors and their associated output pixel position addresses;

calculating an address offset from each output pixel position to each of said temporally adjacent input arrays of input pixel values by multiplying that motion vector associated with said output pixel position by temporal displacement to said input arrays of input pixel values;

adding each of said address offsets to said associated output pixel position address to yield source addresses in each of said temporally adjacent input arrays of input pixel values; and

incrementing said marker values for said source addresses to indicate its use as a source.

11. Apparatus for performing motion compensated image interpolation substantially as hereinbefore described with reference to the

accompanying drawings.

12. A method of performing motion compensated image interpolation substantially as hereinbefore described with reference to the accompanying drawings.

Patents Act 1977
Examiner's report to the Comptroller under Section 17 (The Search Report)

Application number

GB 9313063.1

Relevant Technical fields

(i) UK Cl (Edition L) H4F - FEP, FEX, FGM, FGXX

(ii) Int Cl (Edition 5) H04N - 5/14, 7/01

Search Examiner

D H JONES

Databases (see over)

(i) UK Patent Office

(ii) ONLINE DATABASES: WPI

Date of Search

15 SEPTEMBER 1993

Documents considered relevant following a search in respect of claims 1-12

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
X	GB 2261342 A (BBC) see lines 19-23 page 3, line 11 page 5 to line 21 page 6 and Figure 3	1, 6
X	GB 2259625 A (BBC) see Figures 4 & 8 and lines 14-33 page 5	1, 6
X	GB 2256341 A (BBC) see Figure 2	1, 6

Category	Identity of document and relevant passages	Relevan' to claim,

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family, corresponding document.

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).